The Dissertation Committee for Charles Manuel Lena
certifies that this is the approved version of the following dissertation:

Scalable Electronic Structure Methods to Solve the Kohn-Sham Equation

Committee:

James R. Chelikowsky, Supervisor

Alexander A. Demkov

John G. Ekerdt

Gyeong S. Hwang

Irene M. Gamba

Scalable Electronic Structure Methods to Solve the Kohn-Sham Equation

by

Charles Manuel Lena

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2017

For my Mom and Dad, who continue to support me through trials and tribulations.

For Katie, for both the subtle and the not-so-subtle nudges.

For Tarik, for never falling silent.

For Chris, Paul, Edward, Jim, and my other fast friends.

For the entire crew of Distractions, Inc., the best sanity check a graduate student could have.

For my cat, Tai, who sees both ends.

Acknowledgments

The contained, and many of the related, works depend upon the generous assistance of several agencies, groups, and individuals, including the following:

- My advisor, Dr. James R. Chelikowsky
- Dr. Ariel Biller and Dr. Grady Schofield
- Joshua Neitzel and Kai-Hsin Liou
- The McKetta Department of Chemical Engineering
- Virginia and Ernest Cockrell, Jr. Fellowship In Engineering
- National Science Foundation, Software Grant
- Dept. of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences[1]
- Dept. of Energy, National Exascale Software Application Program

I would like to acknowledge my collaborations with multiple concurrent and previous contributors to our code PARSEC, Pseudopotential Algorithm for

[1]Grant Numbers DE-SC0008877

Real Space Electronic Calculations. There exist several variants of PARSEC, in different languages, tackling different problems within the field of electronic structure problems applied to materials exploration and design. The main variants I reference in this document include the codes and collaborations found in Table 1.1. Many people and groups have contributed to these efforts, some through a checkbook and others through a late night.

Scalable Electronic Structure Methods to Solve the Kohn-Sham Equation

Publication No. _____

Charles Manuel Lena, Ph.D.
The University of Texas at Austin, 2017

Supervisor: James R. Chelikowsky

From the single hydrogen to proteins in the hundreds of thousands of kilodaltons, scientists can use the electronic structure of interacting atoms to predict their material properties. Knowing the material properties through solving the electronic structure problem, would allow for the controlled prediction and corresponding design of materials. The Kohn-Sham equations, based on density functional theory, transform a many-body problem impossible to solve for anything but the smallest molecules, into a practical problem which can be used to predict material properties. Although KSDFT scales as the cube of the number of electrons in the system, there are additional well documented approximations to further reduce the number of electrons, such as the pseudopotential method.

The incoming exascale era will lead to unavoidable challenges in solving the Kohn-Sham equations. These challenges include communication and hardware considerations. Old paradigms, epitomized by repeated series of globally

forced synchronization points, will give way to new breeds of algorithms to maximizing scaling performance while maintaining portability.

This thesis focuses on the solution to Kohn-Sham DFT in real space at scale. Key to this effort is a parallel treatment of numerical elements involving the Rayleigh-Ritz method. At minimum, the Rayleigh-Ritz projection requires a number of distributed matrix vector operations equal to the number of electrons solved for in a system. Furthermore, the projection requires that number, squared and then halved, of dot products. The memory cost for such an algorithm also grows very large quickly, and explicit intelligent management is not an option. I demonstrate the computational requirements for the various steps in solving for the electronic structure problem for both large and small molecular systems. This thesis also discusses opportunities in real space Kohn-Sham DFT to further utilize floating point optimized hardware the with higher order stencils.

Table of Contents

List of Tables

List of Figures

Glossary

?GEMM	appropriately typed complex or real valued general matrix multiply routine
PARSEC	Pseudopotential Algorithms for Real Space Electronic Calculations. Long running real space density functional theory code written mostly in Fortran
ChebFSI	Chebyshev filtering subspace iteration
Cori Phase I	Cori Phase I is a large supercomputer comprised of 2x16-core Intel Xeon Processor E5-2698v3 nodes at the National Energy Research Scientific Computing Center
Cori Phase II	Cori Phase II is a large supercomputer comprised of 68-core Intel Knights Landing and 2x16-core Intel Xeon Processor E5-2698v3 nodes at the National Energy Research Scientific Computing Center
DCMP	Eigendecomposition of the Rayleigh-Ritz quotient matrix returning Ritz values and vectors; fourth step in the Chebyshev Filtering Subspace Iteration
DFT	density functional theory
ELPA	Eigenvalue SoLvers for Petaflop-Applications. Open source highly efficient direct eigensolver library for symmetric matrices. Designed with electronic structure calculations in mind.
FLOP/s	floating point operations per second
FLOPs	floating point operations

FLTR	Chebyshev filtering of the subspace; first step in the Chebyshev Filtering Subspace Iteration
HPC	high performance computing
HSC	Hamann-Schlüter-Chiang pseudopotential
KNL	Intel Knights Landing Xeon Phi architecture
KSDFT	Kohn-Sham density functional theory
matvec	Shorthand for matrix-vector product. Typical representation of sparse Hamiltonian acting on a wave function
MBWE	many-body wave equation
OMP	OpenMP (Open Multi-Processing) provides an API for cross-platform shared memory multi-processing
ORTH	Orthonormalization of the filtered subspace; second step in the Chebyshev Filtering Subspace Iteration
PROJ	Projection of the Hamiltonian onto the filtered subspace to form the Rayleigh-Ritz quotient matrix; third step in the Chebyshev Filtering Subspace Iteration
Rayleigh-Ritz	process for extraction of eigenvalues from a basis spanning a subspace
Stampede2	Stampede2 is a large supercomputer comprised of both 68-core Intel Knights Landing and 48-core Intel Xeon Platinum 8160 "Skylake" nodes at the Texas Advanced Computing Center
TM2	Improved Troullier-Martins pseudopotential
UPDT	Updating the subspace with a Rayleigh-Ritz rotation; final step in the Chebyshev Filtering Subspace Iteration

Symbols

ball of radius r	$B_J(r)$	region of a ball of radius r, centered at coordinates for nucleus J
exchange-correlation potential	V_{xc}	potential corresponding to the choice of the exchange and correlation operators
grid spacing	h	Describes the uniform spacing between points on a given Ω_D. In conjunction with boundary information, determines the number of grid points in the domain. In general, $N \approx O(h^{-3})$. Common range of $h \in [.2, .7]$
Hamiltonian	H	Hamiltonian operator described in DFT. Self-adjoint operator that contains all of the physics for a system. $H \in \mathbb{C}^{N \times N}$, is generally extremely sparse.
Hartree potential	V_H	potential corresponding to the coulombic contribution
ionic potential	V_{ion}	potential corresponding to the pseudopotential contributions
number of grid points	N	Describes the number of grid points on Ω_D where we solve for the wave functions. In conjunction with the number of states, largely determines the amount of memory required for a real space DFT calculation. In general, $N \approx O(h^{-3}), N \gg s$. N does not have a common range
number of states	s	Number of electronic states (eigenpairs) we are interested in from a given Hamiltonian operator. $s \ll N$

set of processors or communicants	P	Set of processing units typically corresponding to MPI ranks.
wave function data	V	$V \in \mathbb{C}^{N \times s}$ is the digital representation of Ψ – when distributed, its actual dimensions are some padded $N_{\text{ldn}} >= N/\lvert\{P\}\rvert$ to guarantee memory alignment.

Chapter 1

Introduction

Scientists and engineers have continually attempted to predict and characterize various materials, utilizing available theories or creating entirely new ones. Density functional theory (DFT) is arguably one of the more impactful theories in the physical sciences in the modern age, having assisted in the understanding of materials. Walter Kohn's and L. Sham's 1964 seminal paper offers a tractable approach to calculating electronic structure [1], and is obviously of great merit. For his development of the density-functional theory, Kohn shared the 1998 Nobel Prize in Chemistry with John A. Pople, who was rewarded for his own development of computational methods in quantum chemistry [2].

To understand the impact of DFT on materials science, we must understand the need for practical methods to predict materials properties. While inherent computational limitations often restrict our efforts to more empirical approaches, materials scientists have always worked towards attaining accurate *ab initio* approaches. Such first principles approaches, lacking empirically defined parameters, capture the underlying physical phenomena rather than the known quantities. For characterizing and understanding materials through

their electronic interactions, solving the Schrödinger's many-body wave equation Equation 1.1 (MBWE) provides incredibly accurate information on the interactions of an entire system.

$$H\Psi = \varepsilon\Psi \tag{1.1}$$

Unfortunately, it leads to intractable problems for all but the smallest molecules. Having the applicable theory but lacking methods of calculation led to the oft-cited epigraph from Dirac [2]. DFT serves as that tractable method, along with pseudopotentials and the Born-Oppenheimer approximation.

> ...It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.
>
> Dirac, 1929 [2]

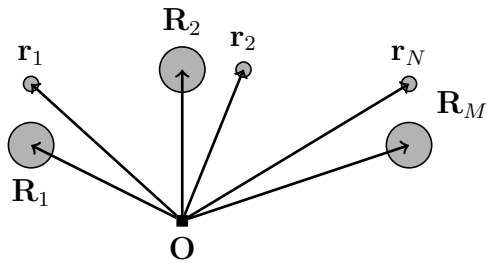

Figure 1.1: An example atomic system.

In the Born-Oppenheimer approximation, the nuclei, (1 to M, at coordinates $\mathbf{R}_J$) in Figure 1.1 are much more massive than the electrons and are

essentially fixed in space. The nuclei coordinates can be treated as classical parameters allowing the separation between the electronic and nuclear wave functions [3].

$$\left(T_e + T_M + U_{MM} + U_{eM} + U_{ee}\right)\Psi = \varepsilon\Psi \tag{1.2}$$

The Born-Oppenheimer approximation justifies the separation of the nuclear portion from the rest of the wave functions, and removes the respective contributions to the Hamiltonian (terms T_{MM} and U_{MM} in Equation 1.2). Solving the remaining electronic terms (1.3)are what most electronic structure methods spend most of their time. In the rest of this work, H and Ψ imply their electronic parts. Several formalisms followed the Born-Oppenheimer approximation, including the Hartree-Fock (1935). Even with a reduction in computational complexity, these formalisms remained largely unused due to remaining computational challenges. The advent of computers changed the ability to tackle the repeated Slater-Determinants seen in the Hartree-Fock.

$$H\Psi = \left[T_e + U_{eM} + U_{ee}\right]\Psi = \varepsilon\Psi \tag{1.3}$$

$$T_e\left(\mathbf{r}^N\right) = -\frac{\hbar^2}{2m}\sum_{i=1}^{N}\nabla_{\mathbf{r}_i}^2 \tag{1.4}$$

$$U_{eM}\left(\mathbf{r}^N, \mathbf{R}^M\right) = -\sum_{K=1}^{M}\sum_{i=1}^{N}\frac{Z_K e^2}{4\pi\epsilon_0|\mathbf{R}_K - \mathbf{r}_i|} \tag{1.5}$$

$$U_{ee}\left(\mathbf{r}^N\right) = \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{e^2}{4\pi\epsilon_0|\mathbf{r}_i - \mathbf{r}_j|} \tag{1.6}$$

Where H Hamiltonian, Ψ wave functions, ε energies, e electron, M a nuclei

Thomas and Fermi independently proposed that the full electronic density is the underlying quantity in the many-body problem for an electron gas. This Thomas-Fermi jellium approximation (or homogenous electron gas) has a fairly severe failing in the handling of the kinetic energy. Over 30 years later, in their 1964 Physical Review paper, Hohenberg and Kohn presented a more rigorous mathematical treatment of the electron density approach [4]. They proved that the electronic density uniquely determines the effective potential v_{eff}, and determines the Hamiltonian H, to within a constant. The electronic density ρ determines the ground state many-body wave function Ψ.

$$\left[-\frac{1}{2}\nabla^2 + v_{tot}(\mathbf{r})\right]\varphi_i(\mathbf{r}) = \varepsilon_i\varphi_i(\mathbf{r}) \tag{1.7}$$

$$v_{tot}(\mathbf{r}) = v_{ion}(\mathbf{r}) + v_H(\mathbf{r}) + v_{xc}(\mathbf{r}) \tag{1.8}$$

$$\rho(\mathbf{r}) = 2\sum_{i}^{N_{occ}} |\varphi_i(\mathbf{r})|^2 \tag{1.9}$$

The next year, Walter Kohn and L. Sham developed density functional theory (DFT) [1]. Within this framework, they introduced the *local density approximation* (LDA), Equation 1.10,

$$E_{xc}^{\mathrm{LDA}}[\rho] = \int \rho(\mathbf{r})\,\epsilon_{xc}[\rho]\,\mathrm{d}\,\mathbf{r} \Rightarrow v_{xc}(\mathbf{r}) = \frac{\delta E_{xc}}{\delta\rho(\mathbf{r})} \tag{1.10}$$

Kohn-Sham DFT is a more tractable framing of the electronic structure problem than the MBWE. Without LDA, these equations are exact solutions to the ground state solutions to the MBWE, but achievable at a fraction of the computational cost. These equations marked the beginning of an *ab initio*

approach to electronic structure calculations. The resulting equations comprise a nonlinear eigenproblem with computational complexity that scales as $O(N^3_{electrons})$, comparing favorably when compared to other quantum methods that can scale as poorly as $O(N^7_e)$.

There are still issues with the computational complexity of the DFT approach. One concern is the number of basis functions needed to span the solution space. The number of basis functions increases linearly with the number of electrons in a system. DFT methods to solve the results generally scaled as $O(N^3_e)$. While far better than the $O(M^K)$ scaling of Schrödinger's equation, even modestly sized molecules like the bucky-ball C_{60} have hundreds of states that require spanning with basis functions. Past the second row in the periodic table, all atoms have at least 10 electrons. This is a difficulty with so called "all electron" calculations. The fight against what is effectively a computational prefactor was already underway.

During the development of DFT, others continued to work on their basis set solutions to the Hartree-Fock approach [5]. There is a very intense focus on finding the next way to reduce the burden of the calculation of such large basis sets. A sort of pseudo-potential was proposed by Hellman in the 1930s and then later Herring, with several others adding to the fray [6], but it is J. Phillips [7] who is often recognized for his empirical pseudopotentials for silicon and germanium. The pseudopotential approximation acts as a method of separating out the tightly bound electrons in the core orbitals while retaining the valence electrons. As valence electrons determine a large portion of bonding

and structural chemistry, they achieve remarkable accuracy. As pseudopotentials reduce the number of electrons requiring consideration, pseudopotentials also achieve a remarkable speedup.

DFT promises ground state properties with the exact density functional. Formulating a first principles pseudopotential theory allows computation of the ground state properties. Hamann, Schlüter, and Chiang introduced a new form of *ab initio* pseudopotentials by the same name (HSC). Their method of construction introduced a *norm-conserving* requirement [8], in which the norm of the all electron and pseudo wave functions to some cutoff radius r_c, as in Equation 1.13. Kleinman and Bylander improved on the efficiency of the norm conserving pseudopotential through a transformation into a projection operator. Projection of the difference between a chosen angular momentum l, called the local component, and all other nonlocal l [9], enables fast calculation of pseudopotential contributions. Pseudopotentials quickly became a strong approximation for electronic structure calculations. There are many different norm conserving recipes, and two other notable formulations are those of Troullier-Martins [10] and Rappe [11]. We tend to use the improved Troullier-Martins pseudopotentials and wave functions, which only have one parameter per angular momentum, a cutoff radius r_c^l.

Since then, scientists have actively used DFT to calculate and examine a multitude of properties around them. A simple search reveals that there exist DFT treatments of nearly every compound that would allow it. In using DFT to calculate a self-consistent electronic structure, we can derive many physi-

cal and electronic quantities, including, but not limited to, interatomic forces, their respective bond strengths, lattice constants, stable structures, transport properties, and band structures. From these quantities we can further explore other related characteristics and methodologies, such as Raman spectra, Atomic Force Microscopy, and Scanning Tunneling Microscopy [12, 13]. With the ability to help explain and debug results from physical experiments on a quantum level, and computational power continuing to grow year after year, DFT enables the analysis of a plethora of characteristics yet unexplained.

As computers continue their increase in speed over the years, and transistor manufacturing processes reach sizes that DFT can actually contend with, we now seem finally to be within reach of calculating not just molecules and infinite crystals, but actual complex systems of interest, such as macro molecules and biological enzymes. DFT is not the only method in the community that achieves verifiable results; but it is one of the cheapest that directly contends with the MBWE and related quantum effects.

1.1 Overview of Research

I contribute to a number of research projects and efforts in the Center for Computational Materials at the University of Texas. This thesis focuses on three major aspects of my work. First, I tender an explanation of the challenges that specifically face DFT and Chebyshev filtering subspace iterations as a method in the era of increasingly parallel and low level computing. Second, I present an in-depth examination of the work in parallelizing the Rayleigh-Ritz

portion of the algorithm, and statistics for PARSEC involving up to tens of thousands of atoms, the current state of the Chebyshev filtering and matrix-vector operator (matvec). These simulations utilized more than 100 million grid points with tens of thousands of states, on both the Intel Knights Landing (KNL) as well as older Haswell systems. Third, I demonstrate and examine higher order stencils for nonlocal contributions on computational complexity in both the matvec representing the Hamiltonian and the post process calculation of forces. I provide visualizations as to the reasons that certain communication avoiding strategies are incompatible with our matvec. I conclude with work done with hodft to demonstrate improved characteristics of molecular dynamics simulations.

1.2 Pseudopotential Real Space Density Functional Theory

Starting from Equation 1.7, we seek to solve the Kohn-Sham equations in real space without the use of a particular basis set. Solving the problem in real space (instead of Fourier space) has specific advantages, discussed by Chelikowsky, Saad, and Troullier [26], when compared to alternative $O(N^3)$ methods like plane waves. There exist inherent communication issues with methods that consistently require Fourier transforms and the resulting collective global communications, which take the form of All-to-All collectives. Real space formulations deal with charged clusters better, and also result in sparse Hamiltonian operators, which can be both a help and hindrance in

today's computational regime. Some physicists dismiss real space electronic structure efforts [27]; the attitude is inconsistent with the success of Hasegawa et al [28, 29] in winning the 2011 ACM Gordon Bell Prize for Sustained Performance Prize. Real space DFT powers scientific applications that push the boundaries of high performance computing (HPC) and electronic structure calculations.

In formulating pseudopotential DFT in real space, we define a regular uniform grid with grid spacing $h = h_x = h_y = h_z < 1$ bohr, such that a set of indices could map to each of the coordinates of the $1..N$ points on the grid. We solve the wave functions Ψ on these grid points. Atoms $1..M$ have specified positions $\mathbf{R}_J$ in the domain, and do not move from the initial setup except in molecular dynamics. We employ a finite difference approximation to represent the Laplacian operator, typically encompassing the 12 neighboring grid points on each of the Cartesian coordinate axes (1.11).

$$\nabla^2\psi\left(x,y,z\right) = \sum_{k=-6}^{6}\sum_{j=-6}^{6}\sum_{i=-6}^{6} C_{ijk}\psi\left(x+i\cdot h, y+j\cdot h, z+k\cdot h\right) \qquad (1.11)$$

We formulate the Hartree potential, v_H as the solution to Poisson's equation (1.12), which we solve through either a conjugate gradient method or multipole expansion, and typically utilize LDA for the exchange-correlation potential v_{xc}.

$$-\nabla^2 v_H(\mathbf{r}) = 4\pi\rho(\mathbf{r}) \qquad (1.12)$$

Occasionally we use generalized gradient approximations (GGA) in place of LDA; but we typically appreciate the computational simplicity of LDA.

1.3 Pseudopotentials

We use Improved Troullier-Martins [10] style pseudopotentials in our DFT algorithms. We are not limited to the Improved Troullier-Martins pseudopotential (TM2) recipe, but it is our standard use, and we have found success with them in predicting even more complex physical interactions, such as atomic force microscopy [12,13]. Pseudopotentials allow for a further reduction in the number of states considered in a given problem through representation of core states as a potential. These potentials replacing the core states acting on the valence electrons drastically reduces computational costs, which again scale with the number of electrons in the system. By definition the pseudopotentials and pseudo wave functions replicate the all-electron eigenvalues in an atomic configuration. Generating pseudo wave functions and pseudopotentials for each l from only the atomic electronic configuration strengthens the claim of *ab initio* methodology. The general nature of pseudo wave functions and pseudopotentials requires us to address the concept of transferability. A pseudopotential's transferability is the characterization of effectiveness of a pseudopotential created in one molecular system achieving correct results in very different physical systems. In the case of the TM2 formulation, the recipe creates pseudopotentials through a full solution of a molecular system consisting of a single atom with the desired electronic configuration. Imposing several requirements, including spherical symmetry of pseudopotential and pseudo wave function and conserving a wave function norm as in Equation 1.13.

$$\int_0^{r_c} |\psi_{PS}(r)|^2 r^2 dr' = \int_0^{r_c} |\psi_{AE}(r)|^2 r^2 dr' \tag{1.13}$$

Pseudopotentials do draw criticisms, however, thanks to their non-uniqueness. Also, pseudopotentials are often coupled with the specific choice of exchange correlation, V_{xc}, used in the pseudopotentials creation, therefore they can inherent the same difficulties in quantifying and optimization. Identifying the underlying qualities of *good* pseudopotentials *a priori* is challenging, but not impossible. On the other hand, identifying what makes a *great* pseudopotential still remains an active area of research in the Center for Computational Materials and elsewhere. Poorly created pseudopotentials, or pseudopotentials extended beyond their limit of transferability, show physical and numerical effects. These deleterious effects include poor convergence for larger structures.

TM2 pseudopotentials have the following basic recipe requirements and prescriptions:

1. The pseudopotential energies ε_{PS} reproduce the all-electron energies ε_{AE} for the elemental atomic in a spherical Schrödinger equation.

2. The pseudo wave functions take the form of the spherically symmetric Equation 1.14.

$$R_l^{PS}(r) = r^l \exp\left(p(r)\right), p(r) = \sum_{k=0,2,..}^{12} c_k r^k \tag{1.14}$$

3. The pseudowave functions are nodeless as well as identical to $R_{AE}(\mathbf{r})$ outside cutoff radius r_c^l.

4. We require their continuity at the cutoff radii $\partial_r^{(i)} R_l^{PS}(r_c^l), i = 0, 1, ..., 4$

5. To qualify as norm-conserving, the TM2 formulation adheres to Equation 1.15

$$2c_0 + \ln\left[\int_0^{r_{cl}} r^{2(l+1)} e\left[2p\left(r\right) - 2c_0\right] dr\right] = \ln\left[\int_0^{r_{cl}} \left|R_l^{AE}\left(r\right)\right|^2 r^2 dr\right] \tag{1.15}$$

At that point, we only need to choose cutoff radii r_c^l and solve the 1D all-electron calculation for a given electronic configuration (e.g. $[1s2, 2s2, 2p6, 3s2, 3p1]$) of an atom. The silicon pseudo wave functions used in the various nanocluster calculations are pictured in Figure 1.2.

While improving pseudopotentials to achieve either faster calculations or more accurate results, another key focus of mine while in the Center for Computational Materials was to improve the calculated properties resulting from potentially subpar pseudopotentials. The pseudopotentials can greatly affect convergence for the system as a whole. The optimization techniques tried included a custom genetic algorithm that trained itself on the transferability of pseudopotentials describing a set of binary III-V compounds, varied convolutions with smoothing functions at cost of accuracy and the use of other expansional forms instead of exponentials for pseudo wave functions. Eventually, these approaches were delayed in favor of utilizing the analytic regularity

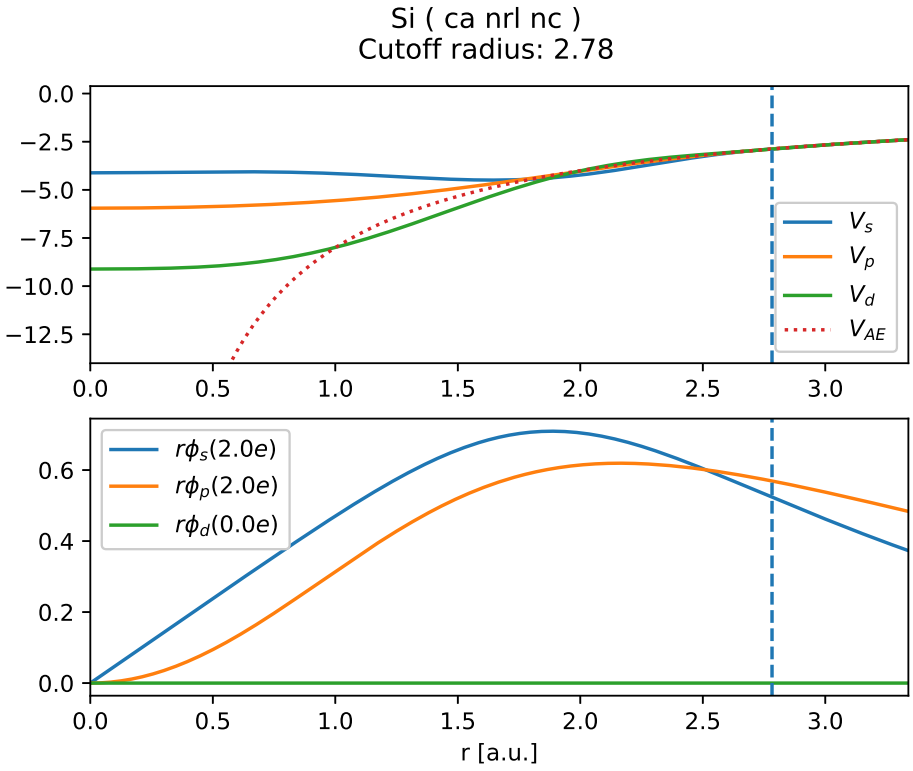

Figure 1.2: Example silicon pseudo wave functions and pseudopotentials from our pseudopotential recipe library used in almost all of the silicon nanoclusters.

of the wave functions taking the form of higher order stencils for more accurate forces, in the prior work of Bobbitt, Schofield, Lena, and Chelikowsky [22]. Through cooperation with J. Neitzel from the Center for Computational Materials I have further expanded on their application to molecular dynamics.

Name	Purpose	Collaborators & References
PARSEC-2.0	Newest Pseudopotential Algorithm for Real Space Electronic Calculations (r2914+), and the focus of this thesis.	Ariel Biller [14], Kevin Gott [15], Kai-Hsin Liou, Intel, Cray, NERSC, TACC
spdft	Specialized real space KSDFT solver that was the heart of Grady's studies into matrix vector and filtering implementation	Grady Schofield [14,16–18]
parsec-1.4	General purpose real space KSDFT solver (r1814+).	Alex Lee, Scotty Bobbit, Yuki Sakai, Masahiro Sakurai, Minjung Kim [14, 17–21]
hodft	Boutique KSDFT solver exploring treatment of pseudopotential contributions	Grady Schofield, Scotty Bobbit, Josh Neitzel [16, 22–25]
PyGA	Python framework for optimization using genetic algorithms. Specialty function included the ability to inform the manager which genes could affect a test. Tested on binary III-V compounds	Grady Schofield [16]
cuda-parsec	Once-off real space DFT solver that utilized CUDA to explore offloaded matrix vector products	

Table 1.1: Specific acknowledgments of code and collaborations. Each of these codes marks a huge effort from multiple collaborators and scientists. More complete statistics, access locations, and revision numbers of these codes to be provided upon request.

Chapter 2

Algorithms at Scale

2.1 Exascale Challenges & Considerations

The computational science community at large has determined the incoming exascale era to be, in a word, unavoidable; various changes will prove challenging to algorithms that for years have avoided considering both communication and hardware restrictions [30]. Old popular paradigms, epitomized by series of fork-joins, are often utilized thanks to their ease of implementation and explanation, but must give way to new breeds of algorithms that strive to both achieve scaling performance while maximizing portability. This chapter focuses on the solution to the real space formulation of Kohn-Sham DFT utilizing filtering at increasingly large computational scale, especially that of the Rayleigh-Ritz steps.

As stated in Chapter 1, the Kohn-Sham equations can be solved through either formulation as an energy minimization problem [31] or as a nonlinear eigenproblem [32]. We choose to solve the problem as a nonlinear eigenproblem, linearized and iterated to a self consistent solution through fixed point iterations. We also focus the analysis strictly on the nonlinear eigenproblem, although thanks to the shared challenges for algorithms in general, we expect

many of the same arguments concerning performance will follow for the energy minimization problem routines.

The overview of forming a self consistent solution from the nonlinear eigenproblem resulting from KSDFT is a relatively straightforward process, depicted in Figure 2.1. While self consistent approaches for solving eigenproblems apply to many more problems than just DFT, this approach works particularly well for DFT thanks to *a priori* knowledge of the solution in the form of the superposition of atomic potentials. Assembling the knowledge of the physical system and initializing the solution as the resulting initial guess typically accelerates the process, as it typically guarantees the fixed-point condition [32].

While direct solvers that employ diagonalization were the preferred solution method for years, the class of so called iterative approaches became increasingly popular thanks to the sparsity of some systems [31]. The main distinction between any direct solver and an iterative solver is the correctness of the solution at a given stage of the process. A direct solver only presents an acceptable solution at the end of the algorithm, whereas the iterative solvers result in acceptable partial solutions much earlier in the process. A well-studied group of iterative approaches includes Krylov subspace filtering methods, with which our Chebyshev polynomial filtering can be compared. In DFT as well as many other eigenproblems, the need for only a small percentage of eigenpairs, a much lower number than those available from the system of interest, drives the applicability of iterative solutions, and has spawned several modern

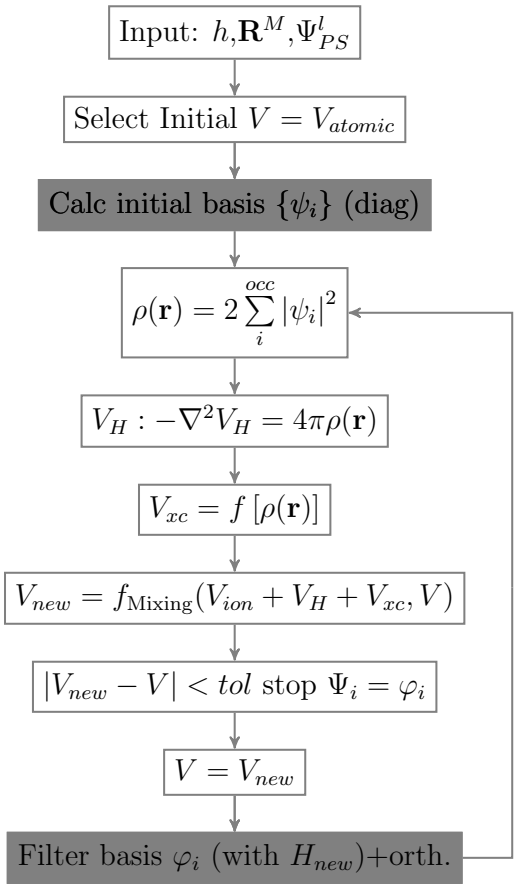

Figure 2.1: Typical approach to a self consistent solution to Kohn-Sham Density Function Theory within parsec-1.4. The initialization of the basis defaulted to a Cheybshev-Davidson approach – and the repeated filtering step marks the subspace iteration.

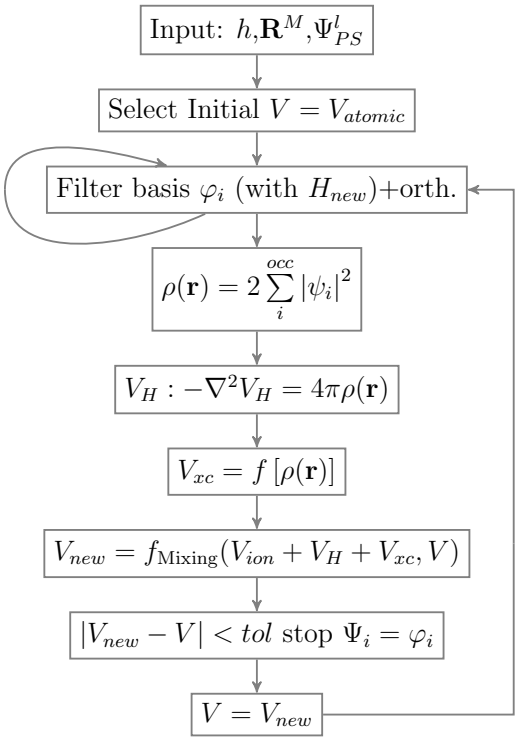

Figure 2.2: Typical approach to a self consistent solution to Kohn-Sham Density Function Theory within parsec-1.4.x – begins with a series of ChebFSI steps holding potentials constant to approximate eigenpairs before updating the potentials and repeating ChebFSI as part of the SCF iteration.

eigendecomposition or diagonalization methods, including specifically Eigenvalue SoLvers for Petaflop-Applications (ELPA) [33, 34], which we utilize for eigendecomposition step within the framework of our iterative solver.

Thanks to the current trends in the evolution of microprocessors [30,35], which favor floating point operations instead of memory throughput, achieving scaling performance with iterative algorithms, or indeed any algorithm based on vectorwise evolution, will require major reworkings. A very simple analysis of the floating point operation rate outpacing the memory bandwidth is achieved through packing the operations together and comparing the time spent on floating point instructions against memory movement. This rudimentary calculation misses the cost of communication, which will be included later in this chapter, but paints a rather bleak picture for typical iterative vector algorithms. This impending requirement affects Chebyshev subspace filtering, and there are several active approaches. Hoemmen [36], Solomonik [37], and Carson [38], working for James Demmel, take analysis of communication requirements a step further in their various formulations of communication avoiding Krylov subspace methods. Our own efforts resulted in a task directed sparse matrix vector operator, which will be outlined in this chapter, important as task directed methods are currently widely considered the best way in handling the enormous numbers of processing units in the exascale era. These advances in the face of challenges give purpose to the results.

When referencing an Exascale supercomputer or cluster, we refer to a massively parallel supercomputer composed as a collection of nodes, of proces-

sors and accelerators, composed of cores, composed of threads, with aggregate simultanous capability of floating point operations per second (FLOP/s) of 10^{18}. I will term a communicant as any of these referenced components that requires communication with any other. The likely configurations of these machines [30] involving millions of communicants, heterogeneous both within a cluster and between clusters, dictate the challenges that current algorithms will face. Namely, the challenges of exascale revolve around dynamic scheduling of interdependent tasks across nodes, avoiding synchronization, and really minimizing communication in general. Waiting for millions of communicants to repeatedly synchronize after reducing and scattering the result is no longer an option for performant algorithms on modern hardware. Similarly, users must utilize the hardware and system extensions at every possible level to ensure that the optimal hardware runs the task at hand.

There are several different manners to attack the most obvious exascale challenges. In spdft, Schofield used surface-volume minimization through boundary sphere partitions to minimize the amount of memory movement required by and during computation. This involved heavy usage of remote direct memory access, which utilizes network interface adaptors ability to access memory to free up the processors from communication tasks. CrayGNI [16] and Infiniband verbs allow implementations to pin memory in place for the communication tasks, avoiding continuously created separate buffers and needless memory movement. Given the size of our dataset with respect to number of grid points, number of states, and the number of atoms, M, (see Table 2.2)

during runs, this is essential to maintaining performance. I furthered the effort through enabling parallel eigendecomposition through ELPA without changing the whole of the extremely custom data structures. It proved challenging, but resulted in self consistent solutions to $Si_{8453}H_{1716}$ (Figure 2.1) without using symmetry reductions, on only 125 Edison nodes at National Energy Research Scientific Computing Center with 1 MPI rank each.

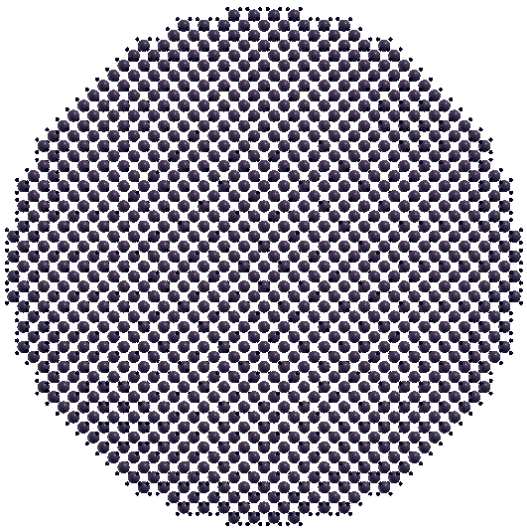

Figure 2.3: Visualization of $Si_{8453}H_{1716}$ nanocluster atomic positions.

While spdft uses a specialty index plane array that reduces memory for large system sizes, PARSEC-1.4.x and PARSEC-2.0 utilize a 3-dimensional Hilbert space filling curve approximation that order the points in the domain as in Figure 2.4. An example when further decomposed across MPI ranks appears as Figure 2.5. This ordering, coupled with a stencil approach, allow for a more readily vectorizing laplacian operator. This ordering and decomposition strategy has some noted benefits as well as some drawbacks [39], but is typically

considered cache optimal when considering only the locality and capacity for vectorization of the laplacian contribution, implemented as stencils, to the Hamiltonian.

$$\begin{aligned} \Psi \quad =: & \begin{pmatrix} \mathbf{V}^0 \\ \mathbf{V}^1 \\ \vdots \\ \mathbf{V}^{P-1} \end{pmatrix}, \mathbf{V}^p = \begin{pmatrix} \Psi_{i_p,1} & \cdots & \Psi_{i_p,s} \\ \vdots & \ddots & \vdots \\ \Psi_{i_{p+1}-1,1} & \cdots & \Psi_{i_{p+1}-1,s} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \\ [N_g, s] := \dim(\mathbf{V}^p) & \end{aligned} \tag{2.1}$$

In describing the grid points, we determine the contiguous data layout. We have kept the parallel distributed $\mathbf{\Psi}$ as $\mathbf{V}$ and distributed along only N, as indicated in Equation 2.1. We should note at this time that $\mathbf{\Psi}$ and therefore $\mathbf{V}$ can be either real or complex valued. Finally, note that a vector in $\mathbf{V}^p$ has some amount of padding to ensure that every first entry in a column $V^p_{1,j} \forall j$ occurs on the aligned memory boundary. This means that basic parallel algebra appears as in Equation 2.2.

$$\begin{aligned} c\Psi_i &= \sum_k^{N_g} cV^p_{ki} \\ \text{dot} = \Psi_i \cdot \Psi_j &= \sum_{p=0}^{P-1} \left(\sum_k^{N_g} V^p_{ki} V^p_{kj} \right) \\ \text{vector}_\alpha = \Psi^T \Psi_j &= \sum_k^{N_g} \Psi_{k\alpha} \Psi_{kj} = \sum_{p=0}^{P-1} \left(\sum_k^{N_g} V^p_{k\alpha} V^p_{kj} \right) \\ \text{matrix}_{ij} = [\Psi W]_{ij} &= \sum_k^{s} \Psi_{ik} W_{kj} = \sum_k^{s} V^p_{ik} W_{kj} \end{aligned} \tag{2.2}$$

Anytime a $\sum_{p=0}^{P-1}$ appears, it indicates the existence of parallel summations in the form of an all reduce, involving necessary messaging and overhead. We note that for the non-distributed matrix W, the operation $X = \mathbf{\Psi}W$ requires no communication, whereas the dot product $X = \mathbf{\Psi} \cdot W = \mathbf{\Psi}^T W$ involves one message of the full result or a subset involving multiple messages. There exists an inherent importance to identifying these operations and scheduling them such that the resulting communications time can spent computing something else. This overlap of communication and computation for algorithms utilizing nonblocking communication is a key requirement for performance at scale [34, 40–43]. When overlap is not possible, current hardware and message overhead prioritizes larger but fewer messages rather than smaller but many more messages.

2.2 Solution to KSDFT without Direct Diagonalization

A general flow for solving KSDFT that we use in parsec-1.4.x is best depicted in Figure 2.1, which corresponds to Algorithm 1. Contrasting a more typical approach depicted in Figure 2.1, we eliminate the diagonalization step in favor of a more approximate Cheybshev filtering replacement. Many DFT algorithms solve step 1.4 with either iterative or direct sparse diagonalization of the Hamiltonian. At scale, this is the mainstay of parsec-1.4.x, resulting from heavy redevelopment of parsec-1.4.x towards parsec-2.0 as we redeveloped approaches that would have a chance of performing well as exascale methods. See Table 1.1 for a brief overview of various approaches and improvements to

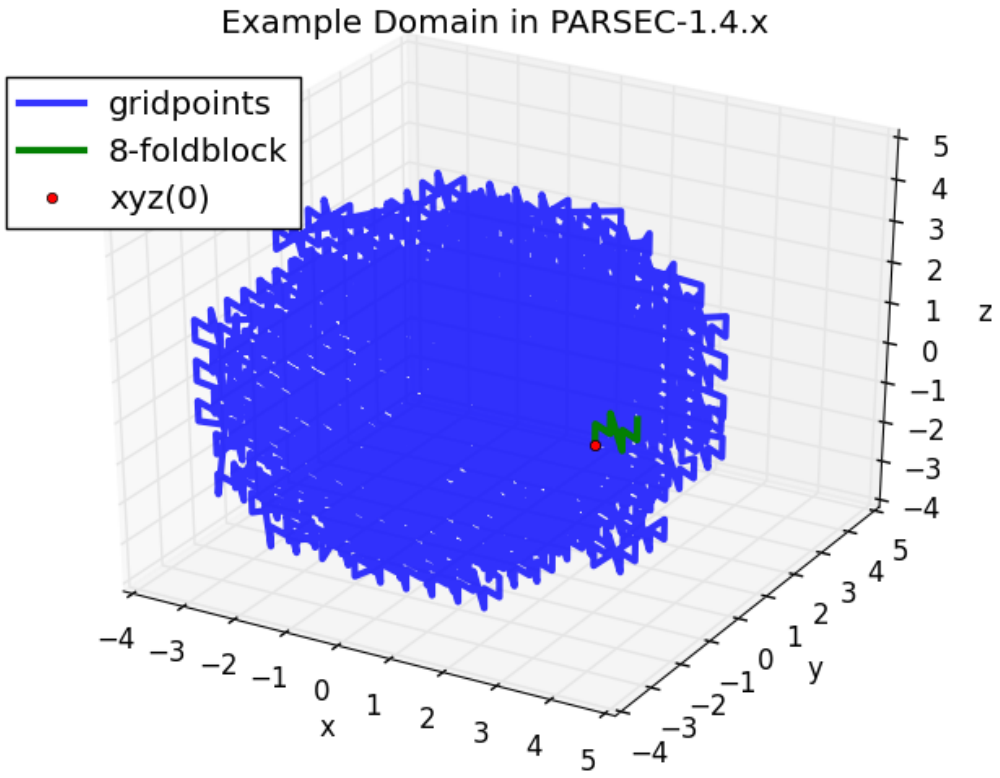

Figure 2.4: Actual space filling curve approximation demonstrating the contiguous traversal in memory across all grid points on parsec-2.0 for $Si_{28}H_{36}$. The focus on utilizing a folding block sized as 8 grid points is to enable vectorization through the use of stencils.

the collection of algorithms and the resulting scientific software packages.

The old 1D parallel distribution of the subspace Ψ across ranks $\{p : p \in P\}$ limits the maximum size of an atomic system. As any atomic system increases in the number of atoms, so too must the number of states also increase in kind; at some size, the further inability to distribute states to share the memory load simply prevents the simulation. Notice that without the dis-

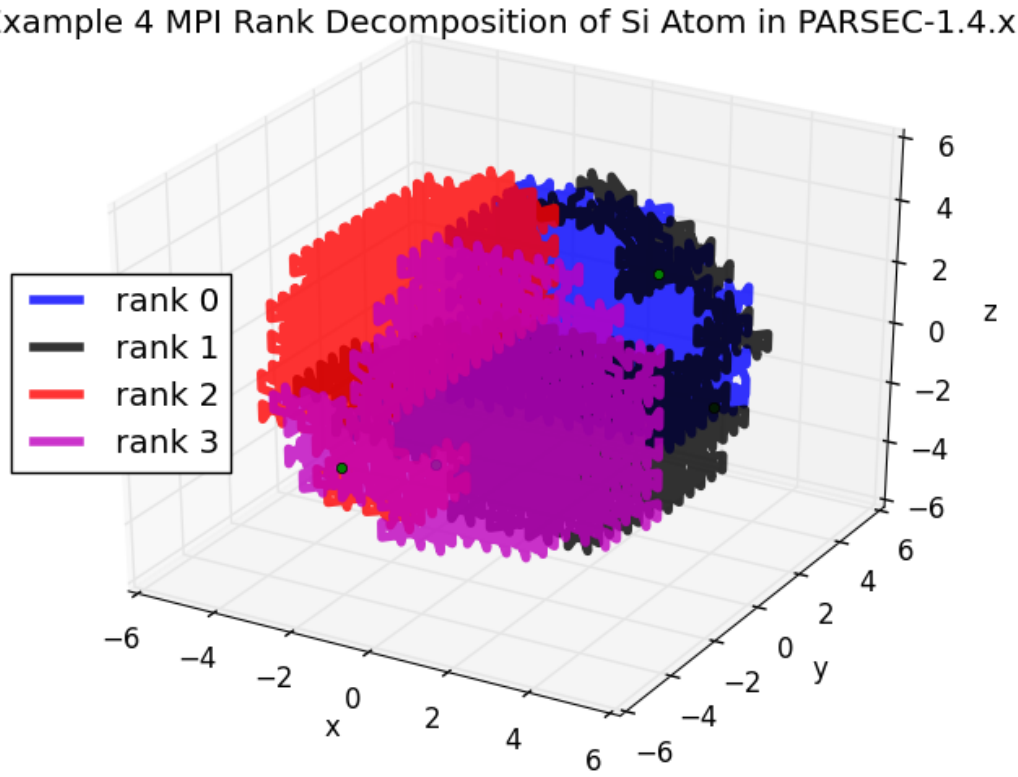

Figure 2.5: Visualizes the distributed partitioning of the SFC approximation across 4 MPI ranks on parsec-2.0 for $Si_{28}H_{36}$, using the same system parameters as in Figure 2.4. The overly small number of grid points forces essentially either a communication of the whole domain, or computation of the whole of the domain prior to communication of the pieces. In either case, timing results are the same – exhibiting better Hamiltonian performance with a single MPI rank.

tributed Rayleigh-Ritz method, the whole of the s^2 dense projection matrix is stored on every MPI rank. Furthermore, parallelizing and distributing the DCMP method through the use of ELPA does not relieve the memory burden; we must address the entire Rayleigh-Ritz method. Even addressing this

Algorithm 1: General overall algorithm. Step 1.4 requires a choice of solution method, and currently defaults to a subspace filtering. N is the number of grid points, s is the number of states, or energy levels, to solve.

Input: Pseudopotentials $V_{l,M_k}^{ps}(\mathbf{r})$ Pseudowavefunctions $\phi_{l,M_k}^{ps}(\mathbf{r})$ Gridspacing h System coordinates $\mathbf{R}_{M_k}$ MPI processes $\{p \in \mathbb{N} | 0 \leq p < P\}$, Energy levels desired s

Output: Self consistent eigenpairs (λ_i, ψ_i), occupations occ_i

1.1 $\Omega_D, N, \mathbf{R}', ... \leftarrow f(h, P, \mathbf{R}, ...)$; `// setup grid,domain,shifts`

1.2 $\rho^{\{0\}}(\mathbf{r}), V_{\text{total}}^{\{0\}}(\mathbf{r}) \leftarrow$ superposition of pseudo-quantities;

1.3 $\mathbf{\Psi}^{\{0\}} \leftarrow$ random number generator;

for $\kappa = 1$ **to** $\kappa_{\max}$ **do**

1.4 $\quad \mathbf{\Psi}^{\{\kappa\}}(\mathbf{r}) \leftarrow H[\rho^{\{\kappa-1\}}(\mathbf{r})]\mathbf{\Psi}^{\{\kappa\}}(\mathbf{r}) = E^{\kappa}\mathbf{\Psi}^{\{\kappa\}}(\mathbf{r})$

1.5 $\quad \rho^{\{\kappa\}}(\mathbf{r}) = \sum_{i=1}^{s} occ(i) |\mathbf{\Psi}_i^{\{\kappa\}}(\mathbf{r})|^2$

1.6 $\quad V_H^{\{\kappa\}}(\mathbf{r}) \leftarrow \Delta V_H(\mathbf{r}) = 8\pi\rho^{\{\kappa\}}(\mathbf{r})$

1.7 $\quad V_{xc}^{\{\kappa\}}(\mathbf{r}) \leftarrow V_{xc}[\rho^{\{\kappa\}}(\mathbf{r})]$

1.8 $\quad V_{\text{total}}^{\{\kappa\}}(\mathbf{r}) = f_{\text{mix}}(V_{xc}^{\{\kappa\}}(\mathbf{r}) + V_H^{\{\kappa\}}(\mathbf{r}) + V_{ion}^{\{0\}}(\mathbf{r}), V_{\text{total}}^{\{\kappa-1\}}(\mathbf{r}))$

1.9 $\quad$ Check convergence

end

method cannot preclude a change in how the subspace must be distributed in the future versions of parsec – while some states can be mirrored on each process, eventually the processes lose the ability to hold every state for any meaningful subset of grid points. However, using the rough memory estimate of Equation 2.13, we can examine feasible values – and memory would not prevent the 1D distribution from running until s is nearly 400,000, assuming a node count of around 9000, and a N/P ratio of 5000. Instead, it would be synchronization between those nodes that would bring the algorithm to a halt.

The problem converges through self-consistent (SCF) iterations, and these should make up the bulk of the time. Each Chebyshev filtering SCF iteration can be decomposed into 5 operations – Chebyshev filtering (FLTR), orthonormalization (ORTH), and a Rayleigh-Ritz method, which is itself 3 steps – projection (PROJ), eigendecomposition (DCMP), and update (UPDT). The Rayleigh-Ritz method provides resultant eigenvalues and refinement vectors to our subspace, also known as Ritz values and Ritz vectors [44]. Analyzing and deciding on an efficient effective parallelized approach for these three steps – within the confines of our specific pseudopotential real space DFT algorithm and its prior discretization is a very important interdisciplinary task so as not to continue the 20 years of research and physics formulation. It is the major topic of this thesis. See the following Chapter 3 for more detail on the design process.

2.2.1 Chebyshev Filtering - FLTR

Chebyshev filtering subspace iteration (ChebFSI) utilizes Chebyshev polynomials to emphasize selectively windows of eigenvalues. We tailor this process to ensure that found eigenpairs correspond to the occupied states with some buffer of unoccupied states, eliminating the overhead of solving for every state within the Hamiltonian. Zhou [32, 45] introduced this in 2006, and improved upon the process in a later collaboration with the Center for Computational Materials [46]. This lead to the Chebyshev filtering first method, [47] which allows the DFT algorithm to entirely avoid a typical iterative diagonal-

ization, replacing it with 3 to 4 steps of ChebFSI without updating the initial charge density, derived from the super position of atomic orbitals. The process produces enough of a quality set of eigenpairs to continue the typical ChebFSI iterations towards self consistency.

When implementing the actual filtering stage, we at the Center for Computational Materials, and others who replicated our efforts [41, 43], implement Chebyshev filtering operations as a three term recurrence, as in (2.3). Note that its corresponding dense algebra would naturally form a matrix-matrix product with multiple states considered – but the Hamiltonian is typically an extremely sparse operator (see Section 2.3), and so in the past we have implemented it as a highly performant matrix-vector product [16]. The performance of this matvec operator often determines the ability of much of the algorithm to scale, especially as FLTR calls it m times to apply the polynomial $p_m(H)$ to each state, and to a lesser extent PROJ which calls it once per computed states. The operator itself involves a number of nontrivial contributions in the form of the pseudopotential-derived V_{ion}; its difficulties are explored in a later chapter.

$$\mathbf{V}_{k+1} = \frac{2\sigma_{k+1}}{e} \left(\mathbf{H} - c\mathbf{I}\right) \mathbf{V}_k - \sigma_{k+1}\sigma_k \mathbf{V}_{k-1} \tag{2.3}$$

Ongoing work exists on a fully blocked-layout Chebyshev filtering. The matrix formulation is obvious when looking at (2.3) – but until this time a strong emphasis was placed on the matrix-vector product, which only operates on a single vector at a time. In the past this style of work was very

effective, but as cores become more numerous, yet weaker in terms of clock speed, while having higher penalties for accessing memory out of cache, and memory bandwidth that has not kept pace with increases in FLOP/s, the efficacy is fading. To best change with the hardware, this new matrix operator for the Hamiltonian requires a modified matvec communications cycle that can utilize blocks of vectors in proper matrix-matrix operations, instead of pure matrix vector operations. In that regard, we need to develop matrix-matrix formulations of the computational work and algorithms when available. This necessity for cache coherency forms a driving force behind the matrix-matrix operations, which are the most easily achieved, and portable, vehicle there. We also continuing the advances in task based operations that make the current matvec used within Chebyshev filtering operation in parsec-1.4.x, which is successful at attaining hyperthreaded performance gains (see Table 3.1).

2.2.2 Orthogonalization - ORTH

Orthogonalization provides numerical stability in the face of repeated filtering operations, and is also a requirement for the Rayleigh-Ritz method, which a basisless DFT solution requires. Block orthogonalization in most literature refers to the projection for one vector against blocks of vectors happening at once, as in Equation 2.4, proceeding from the variants of the Gram-Schmidt methods. Orthonormalization is a common process shared by more than DFT methods, and so more performant orthogonalization methods are an active area of research, including the communication avoiding methods of Solomonik

et al [37] and task based Singular Value Decomposition (SVD) methods of Haidar et al [48]. The current utilized DGKS method and iterated classical Gram-Schmidt (2.4) will hopefully give way to one of these in the future.

$$\psi_k^{\perp} = \psi_k - \sum_{j=1}^{k-1} \frac{< \psi_j^{\perp}, \psi_k >}{< \psi_j^{\perp}, \psi_j^{\perp} >} \psi_j^{\perp} \tag{2.4}$$

2.2.3 Rayleigh-Ritz Projection - PROJ

The Rayleigh-Ritz projection step takes an orthonormalized subspace and calculates the projection of the sparse operator representative of the Hamiltonian of a given system in DFT on that subspace to form the Rayleigh quotient matrix $\mathbf{G} \in \mathbb{C}^{s \times s}$. The pure algebraic representation of the projection operation is Equation 2.5.

$$\mathbf{G}_{ij} = \mathbf{\Psi}_i^T \mathbf{H} \mathbf{\Psi}_j \tag{2.5}$$

We see that there are a few manners of expressing the cost in so far as FLOPs – the most straightforward being s matvecs + $\frac{s(s+1)}{2}$ dot products of length N. The challenges in scaling PROJ mostly revolve around balancing bandwidth and throughput. On light weight cores, like those used in the KNL, it becomes increasingly important to arrange as many of the sub operations in matrix representations as possible – this will be discussed in the Rayleigh-Ritz development section.

2.2.4 Rayleigh-Ritz Decomposition - DCMP

After construction of the Rayleigh quotient matrix $\mathbf{G} \in \mathbb{C}^{s \times s}$, we need to solve the eigenproblem associated with Equation 2.6. We do this with ELPA when accessible, which solves the eigendecomposition through the use of a three stage process [33, 34], and has been shown to scale to problem sizes in excess of what we are anticipating for this generation of problems.

$$\mathbf{G}\mathbf{W}_i = \lambda_i \mathbf{W}_i \tag{2.6}$$

Utilizing ELPA-2014 allowed for a fully self consistent $Si_{8453}H_{1715}$ without symmetry on spdft. The major drawbacks and influential cascading design decisions for both spdft and parsec-1.4.x were twofold – using ELPA forces use of BLACS and ScaLAPACK, as well as the required full s^2 storage of $\mathbf{G}$ instead of an alternative.

ELPA makes use of the same block cyclic distribution that ScaLAPACK uses. Thanks to the fact that parsec-1.4+ distributes Ψ across MPI ranks only across N and not s, and does not do so in a block cyclic manner, there exists contention for functions mapping the 2D block cyclic distributed objects back to 1D distributed objects. The use of BLACS means that we lose explicit control over allocation of communication buffers, which can form a source of lost computational hours when BLACS crashes large simulations with undetectable bad allocations. We manage this by polling the GNU/Linux statm page throughout PARSEC's runtime and allocation process and changing operations accordingly. This is most notable in the following step, UPDT.

Furthermore, even though $\mathbf{G}$ is also Hermitian thanks to H being Hermitian, and could be stored with only $s(s+1)/2$ elements, the full matrix must be stored for ELPA, which involves some additional upfront memory movement. $\mathbf{G}$ and $\mathbf{W}$ are the only two block cyclic distributed matrices, and in this DCMP step the resulting $\mathbf{W}$ overwrite $\mathbf{G}$.

2.2.5 Rayleigh-Ritz Update - UPDT

The Rayleigh-Ritz update step utilizes the Ritz vectors from the DCMP operation to refine and correct the evolving subspace. Algebraically, the update step is best understood as a dense matrix matrix operation depicted in (2.7). While simple on its face, the complexity of UPDT quickly increases from any parallel decomposition. In a case where $\mathbf{W}$ and $\mathbf{\Psi}$ are not localized in the same pattern, or even distributed at all, the transmission of remote data and increased complexity of navigating the switches quickly increases the amount of time to solution.

$$\mathbf{\Psi} = \mathbf{\Psi}\mathbf{W} \tag{2.7}$$

This spells trouble in our case, as the different storage schemes for $\mathbf{W}$ and $\mathbf{\Psi}$ means utilizing established parallel general matrix multiply routines, like the one from Parallel BLAS, are generally costly and inefficient due to data transformations. As a dense matrix matrix operation in parallel, each level of required communication requires some amount of additional overhead.

2.3 Classic Matvec Algorithm

It should be evident that the performance of the Hamiltonian operator, represented as a matvec, is integral to the algorithm's overall performance at any computing scale. The reader should be reminded that the full Hamiltonian operator contains all of the physics behavior desired, and it must interact with the knowledge of the domain. These features make any given representative matvec a very custom implementation to any DFT codebase utilizing subspace iteration solvers.

A matrix operator representation of the Hamiltonian for the Kohn-Sham equations is fairly simple. The full matrix itself should never be stored, not even as a full sparse representation, as the largest number of entries in any 0D or 1D problem will be the same repeating finite difference coefficients which can be represented through a series of Kronecker products. Figure 4.3 demonstrates the structure of a visualized Hamiltonian and the resulting fill pattern as evidence. Parsec 1.4.x stores the nonlocal contributions as a compressed sparse row matrix. As noted, there are only two sources of off diagonal entries in the real space DFT Hamiltonian. First, the finite difference approximation of the laplacian causes a dependency on grid points to the 12 neighbors on each of the Cartesian coordinates. Second, we can separate V_{ion} as in Equation 2.9. V_{nloc} produces dependencies for various grid points on surrounding neighbors, depending on their positioning relative to the M atoms. The Kleinman-Bylander projection formulation allows for a much faster evaluation of pseudopotentials; however, it comes at the cost of an integration in

the ball of radius r_c^l, $B_J(r_c^l)$, around each atom as in Equation 2.8.

$$v_{ion,J} = v_{loc,J} + \sum_{J,l,m} \frac{|v_l^{'}(r)\phi_l(r)Y_{l,m}(\theta,\varphi)\rangle\langle Y_{l,m}(\theta,\varphi)\phi_l(r)V_l'(r)|}{\langle\phi_l(r)|V_l'(r)|\phi_l(r)\rangle} \tag{2.8}$$

$$V_{ion}(r) = \sum_{J}^{M} \begin{cases} r_J \leq r_c^l & v_{loc,J} + v_{nloc,J} \\ r_J > r_c^l & -\frac{Z}{r_J} \end{cases} \tag{2.9}$$

$$v_{nloc,J}\mathbf{\Psi}_i = \sum_{l,m,J} \frac{|v_l^{'}(r)\phi_l(r)Y_{l,m}(\theta,\varphi)\rangle\langle Y_{l,m}(\theta,\varphi)\phi_l(r)V_l'(r)|\mathbf{\Psi}_i\rangle}{\langle\phi_l(r)|V_l'(r)|\phi_l(r)\rangle} \tag{2.10}$$

$$= \sum_{l,m,J} \frac{|v_l^{'}(r)\phi_l(r)Y_{l,m}(\theta,\varphi)\rangle \int\limits_{B_J(r_c^l)} Y_{l,m}(\theta,\varphi)\phi_l(r)V_l'(r)\mathbf{\Psi}_i dV}{\langle\phi_l(r)|V_l'(r)|\phi_l(r)\rangle} \tag{2.11}$$

With this knowledge in hand, we can address the operations count for the representative operator for the Hamiltonian, currently expressed as a custom matvec operator. It is hard to achieve an absolute count on the number of FLOP/s a matrix vector operator requires, but an estimation better than $O(N)$ based on the terms is quite reasonable:

$$\Theta_{\text{matvec}} \approx \left(\underbrace{74}_{FD} + \underbrace{1+1}_{V_H, V_{xc}} + \overbrace{1}^{\sum\limits^{M} V_{loc,J}} \right) N + \underbrace{\left(\frac{4\pi}{3} \left(\frac{r_c^l}{h} + 1 \right)^3 M \right)^2}_{\approx \#\{xyz\} \subset B_J(r_c^l) \forall J} \tag{2.12}$$

In estimating the terms due to the nonlocal potential, we notice that there are two major terms to consider: $M << N$ for small systems, but typically $s/10 \leq$

$M \leq s$, which means the handling for the nonlocal potential contributions absolutely must be optimized within the rest of the framework.

Theoretical memory scaling of the various steps can now be adjusted per step without distributing the entire process. We can actually reduce runtime significantly at the cost of duplicate data storage, see Table 2.2, during certain operations. This eventually becomes memory cost prohibitive for high enough counts of nodes Equation 2.13,

$$Memory \geq 2Ns\frac{|\{p\}|}{|\{P\}|} + s^2\left(1 + \frac{|\{p\}|}{|\{P\}|}\right) + s \tag{2.13}$$

but until then can be exploited and even worsened in a memory sense to exhibit gains in computational performance during the PROJ and UPDT. In this vein, some of the updated operations attempt to maximize usage of available system resources through monitoring calls to the system kernel, changing allocations for buffers and temporaries accordingly.

2.4 Conclusions

We explain the algorithm, comprised of Cheybshev filtering subspace iterations, that we use in the pseudopotential real space Kohn-Sham DFT formulation. We explain the importance of the Hamiltonian operator in the formulation, and our estimations as to each invocation's computational cost. Analyzing the computational cost of the Hamiltonian operator, in addition to the defined parallel distributed algebra for operations on the set of 1D-distributed wave functions, we explain the requirements for each of the five

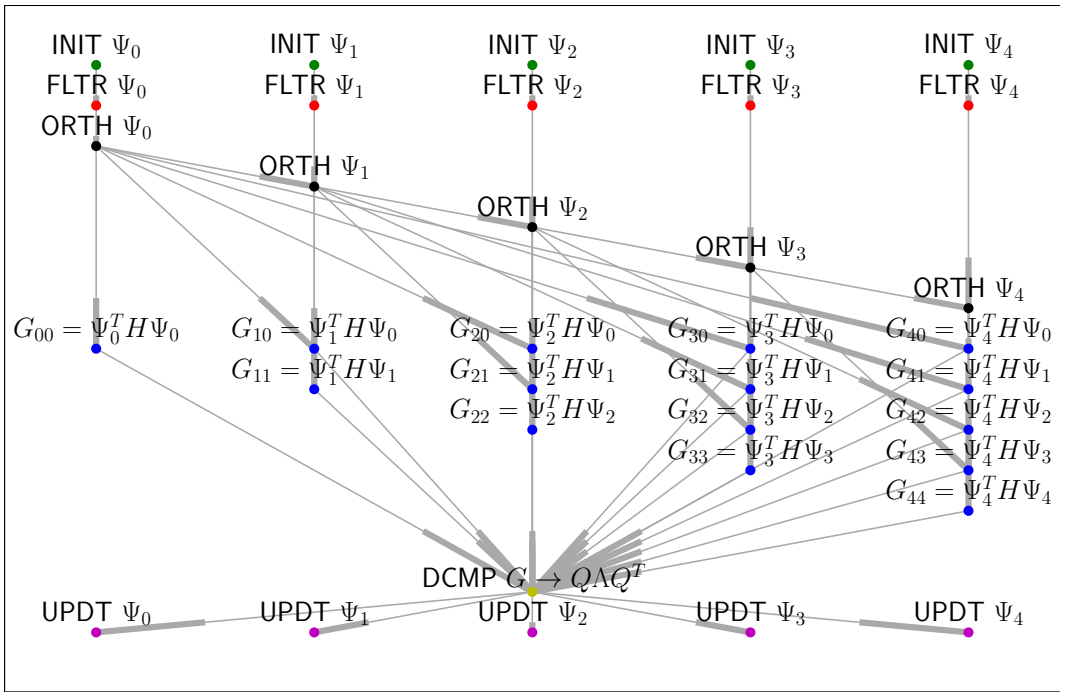

Figure 2.6: High level overview of the data dependencies between the 5 major steps of a Cheybshev filtering subspace iteration in real space DFT. Lines depict dependencies, with the blocks indicating the dependent. It should be noted that the FLTR operations are entirely independent of each other, and that the eigendecomposition – DCMP – is a major sync point as it both requires an entire quotient matrix **G** and passes it to an external library.

major steps in a Chebyshev filtering subspace iteration, including the memory memory requirements and expected FLOP counts.

We demonstrate claims about single node hyperthreading use with a small silicon nanocluster, $Si_{28}H_{36}$. We then show multinode timings and expected scaling, based on initial timings, for a much larger silicon nanocluster $Si_{3917}H_{1036}$ with regards to changes in N and s. Recognizing that new hardware is very cache conscious and memory limited, we suggest optimizing cer-

tain aspects of the algorithm as matrix-matrix operations, utilizing buffers to gather results as to prefer larger and fewer matrix-matrix operations in place of matrix-vector or vector-vector.

Algorithm 2: Scaled Chebyshev filtering algorithm from PARSEC.

Input: $V \in \mathbb{C}^{N \times s}$, $H \in \mathbb{C}^{N \times N}$, Chebyshev polynomial order m, lowest bound a, lower bound a_L, upper bound b

Output: Filtered V

allocate blk comm. buffers `// ` $a_L = a$ `in nonscaled vers.`
$n_{blk} = s/blk$; $\kappa(1:m) = n_{blk}/m$; $e = (b-a)/2$; $e_{rp} = 2/e$;
$c = (b+a)/2$; $\sigma = e/(c - a_L)$; $\sigma_1 = \sigma$;
$\sigma_{ei} = \sigma_1/e$; $\sigma_2 = 0$; vk = 1 - blk;
Aligned fastmem allocate $V_{\{1\}}, V_{\{2\}}, V_{\{3\}} \in \mathbb{C}^{N \times blk}$;
do k_m = *1,m*
  do $\kappa_{block} = 1$**:** κ_{k_m}
    vk = vk + blk; iteratively prime blk buffers
    $V_{\{1\}} = V_{vk:vk+blk-1}$;
    $V_{\{2\}} = H(V_{\{1\}}, \text{buffers})$;
    $V_{\{2\}} = (V_{\{2\}} - c * V_{\{1\}}) * \sigma_{ei}$;
    do i = *2, pm(*k_m*)* `// poly degree per block`
      iteratively prime blk buffers;
      $V_{\{3\}} = H(V_{\{2\}}, \text{buffers})$; $\sigma_2 = 1/(2/\sigma_1 - \sigma)$;
      $V_{\{3\}} = (V_{\{3\}} - c * V_{\{2\}}) * e_{rp}$;
      $V_{\{3\}} = V_{\{3\}} - \sigma * V_{\{1\}}$;
      $V_{\{1\}} = V_{\{2\}}$;
      $V_{\{2\}} = V_{\{3\}} * \sigma_2$;
      $\sigma = \sigma_2$;
    $V_{vk:vk+blk-1} = V_{\{2\}}$
  $sigma = e/(c - a_L)$
3.1 cleanup buffers;

Stampede2	Data: Filter [sec]					
Si28H36	MPI	TOTAL HW THREADS			SPEEDUP	
Hamil. Rank	RANKS	68	136	272	2 HT	4 HT
855568	1	11.55	8.52	7.63	1.36	1.51
	2	15.33	15.1	17.45	1.02	0.88
	4	16.04	16.39	19.45	0.98	0.82
1575616	1	21.52	15.88	14.18	1.36	1.52
	2	27.83	25.64	28.165	1.09	0.99
	4	27.11	25.915	29.225	1.05	0.93

Table 2.1: Demonstrates hyperthreading performance for the FLTR step on a single KNL node from National Energy Research Scientific Computing Center's Cori, for $Si_{28}H_{36}$. Of note is the strong scaling resulting from the utilization of additional hyperthreads, if and only if there exists enough work to provide to them.

Operation	Total FLOPs	max additional memory
FLTR	$ms\Theta_{\text{matvec}}$	$3N\kappa_{blk,FLTR} + O(n_{buffer})$
ORTH	$O(Ns^2)$	$N(s+1)+s$
PROJ	$Ns(s+1)/2 + s\Theta_{\text{matvec}}$	$\kappa_{\text{matvec}}\kappa_{blk,PROJ} + s^2$
DCMP	$O(s^3)$	$O(f(s,P)) + s^2$
UPDT	$O(Ns^2)$	$Ns + s^2$

Table 2.2: Statistics on the 5 major steps of a Chebyshev filtering subspace iteration. We assume that 'additional memory' does not include at least the sized Ns global sized Ψ which must exist at all times – so additional memory See (2.12) for Θ_{matvec}

Molecule	N [1e6]	s [1e4]	FLTR	ORTH	PROJ	DCMP	UPDT
$Si_{3917}H_{1036}$	1.45	1	407.25	289.68	200.63	8.14	107.66
$Si_{3917}H_{1036}$	2.93	1	588.16	643.26	410.42	8.63	195.21
$Si_{3917}H_{1036}$	1.45	1.5	613.63	953.21	503.01	16.03	222.15
$Si_{3917}H_{1036}$	2.93	1.5	878.22	2038.9	1278.3	16.3	442.04

Table 2.3: Early distributed timing results in seconds for $Si_{3917}H_{1036}$ nanocluster of the five various stages that comprise an iteration of subspace filtering. All cases were run on 16 Intel KNL nodes at TACC, using quad cache mode and a 4x17 hybrid MPI+OpenMP configuration. This is running fully distributed, untuned version of all pieces of parsec-1.4.x. Tuning for the hardware (e.g. the KNL) makes a huge difference, as demonstrated later on PROJ applied to an Insulin hexamer.

Chapter 3

Distributed Rayleigh Ritz Algorithm

3.1 Considerations & Algorithm

The Rayleigh Ritz method is a fundamentally straightforward method Algorithm 3, easily defined in three major supersteps: formation of the Rayleigh quotient matrix, $\mathbf{G}$, through projection, eigendecomposition of $\mathbf{G}$, and update of the basis vectors with the resulting eigenvectors of $\mathbf{G}$ [16, 31]. We will refer to these steps throughout this section as projection, decomposition, and update.

Algorithm 3: The Rayleigh-Ritz method

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$

Output: Approximate eigenpairs (λ_i, Ψ_i)

3.1 $\mathbf{G}_{ij} = \mathbf{V}_i^{\mathrm{H}} H \mathbf{V}_j$; `// form Rayleigh quotient matrix,` $O(Ns^2)$

3.2 $\mathbf{W}^{\mathrm{H}} \Lambda \mathbf{W} = \mathbf{G}$; `// diagonalize` G, $O(s^3)$

3.3 $\Psi = \mathbf{V}\mathbf{W}$; `// form Ritz vectors,` $O(Ns^2)$

Once again, the name of the game in taking the old approaches and rethinking the algorithms for performance is to use as many matrix-matrix operations as possible. In some cases, most notably the exploration of tiled decompositions, doing additional unneccessary work achieves large speedups.

This speedup, in addition to the unused work, usually comes with the requirement of additional memory.

At the same time, the desire to utilize libraries external to our group creates additional requirements. Specifically, we desired a parallel solution to the eigendecomposition and update steps that would not create a large dependency chains of libraries and setup steps. They also needed to provide Fortran friendly interfaces. While there are now many solutions provided for these problems, thanks to the previous requirements we decided on ELPA [33, 34]. The update step currently uses pieces of ScaLAPACK [49] thanks to the pre-existing challenges of fitting to the data structures of parsec-1.4.

While we only apply the Rayleigh Ritz to KSDFT, this algorithm has a myriad of applications [31]. The author expects the lessons learned and presented in this section will apply to many of these other applications that seek to find performance on newer hardware. The reader should note that their performance gain will be highly dependent on their commitment to pre-existing data structures and the decompositions across parallel groups therein.

Specifically, while our parallel decomposition of $\mathbf{V}$ along only grid points, and not states, will ultimately limit scalability of the full problem thanks to memory requirements, it does allow for both the projection operator and update operator to avoid communication during the most floating point intensive sections between eigenstates. Avoiding or hiding this communication is essential to maintaining performance - but the science requirements of truly massive structures as discussed in earlier sections demand adaption

to full parallel data decompositions. These effects have been explored specifically within the confines of single operations by Solomonik, Carson, Knight and Demmel [37]. In previous work, we have explored the use of spectrum slicing in order to deal with large numbers of eigenstates in a manner that would decompose along the states s. We will point out how important this can be during the exploration of the algorithms and muse how to adapt future work for data decompositions along both axes.

3.1.1 Projection

The projection step, Algorithm 3.1, is perhaps the important superstep of the overarching algorithm in terms of the cascading effects of decisions regarding distribution of both memory layout and computational order. The decisions made for the algorithm here with regards to $\mathbf{G}$ affect the eigendecomposition, insofar as the method of calculation directly as well as possible efficiency. Thanks to the algorithm's input vector basis $\mathbf{V}$ having its own predetermined data decomposition and layout, and in the case of parsec-2.0, only being decomposed along the grid points and not the states, we find ourselves locked into a smaller set of options with acceptable changes.

We depict the first method, the most naive method, in Algorithm 4. Interestingly enough, it is the result of years of modifications by various users of parsec-1.4 who, having hit various boundaries of their own systems with regards to memory or buffers, and not focused on the Rayleigh Ritz algorithm itself, instituted various changes of their own. The resulting algorithm is

nonperformant for anything but the smallest systems (as quantitized by s), yet remains an interesting result for the metastudy of active research codes where science, not development, is the primary focus.

Proj. Label (Algo)	# Matvecs	# $\sum_P \mathbf{G}_{\text{sub}}$	# buffers	Mult. Method
Naive Base(4)	s	s	1*	DDOT
Classic(5)	s	1	2	DGEMV
Classic Tiled(6)	s	1	2+	DGEMM

Table 3.1: Table of categorizations and possible quantitizations of projection refactorizations resulting from the exploratory activities. All experiments were performed on the parsec-1.4 branch elpa-simple branch at r1600+ branch or derivative branches therein. Note that the Naive Base from parsec-1.3 used an entirely different, and monolithic, communication scheme.

The naive base implementation within parsec-1.3 was extremely simple, and is hard to even compare line-for-line to parsec-1.4.x thanks to a complete reworking of the matrix vector product. Based on the mapping, we can see that only internal buffers were used, and the intention to only transmit the necessary $s^2/2$ pieces of the symmetric $\mathbf{G}$ translated into the use of s synchronizing global summations – one for each wave function.

3.1.2 Decomposition

Decomposition solves the eigenpairs $(\lambda_i, \mathbf{W}_i)$ of the Rayleigh quotient matrix. Given the aforementioned regional spectral nature of the problem, any eigendecomposition method used in a massively parallel simulation should support solving for only the lowest fractional percentage of the total number

Algorithm 4: The naive projection algorithm.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$

Output: Rayleigh projection matrix $\mathbf{G}$

```
for j = 1 to s do
    u = H V_{:,j}
    for i = j to s do
        G_{ij} = V_{:,i} · u
    end
end
for j = 1 to s do
    global sum G^p_{j:s,j} over all MPI processes p ∈ P
end
```

of eigenvalues. This allows the algorithm to spend computational power on the solution of the bound electrons instead of potentially wasting time on undesired unbound states.

To this end, and because it offered the lightest dependency chain for data structures, we decided on using ELPA [33, 34], then a two stage solver for parallel eigendecomposition, on both spdft and parsec-1.4.x. We retained the nondistributed LAPACK option for compatibility. There are several alternatives for eigendecomposition, but most are only a component of a larger ecosystem like Trilinos [42] or Elemental [50]. We deemed the effort in adapting these entire ecosystem while maintaining the existing physics features of parsec-1.4 to be overwhelming for a research group that often has only one member focused on programming. The challenge of writing and maintaining research code while outputting science in the parallel computing era should not

Algorithm 5: The classic projection with a single global summation.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$

Output: Rayleigh projection matrix $\mathbf{G}$

setup 2 initial buffers;
work $\mathbf{u} \in \mathbb{C}^{N/P}$;
for $j = 1$ **to** s **do**
    $\mathbf{u} = H\mathbf{V}_{:,j}$
    start comms. for $H\mathbf{V}_{:,j+1}$
    $\mathbf{G}_{j:s,j} = \mathbf{V}^{H}_{:,j:s}\mathbf{u}$
end
global sum $\mathbf{G}$ over all MPI processes

be understated, and the Department of Energy provides resources in research efforts, like the NERSC Exascale Scientific Application Program, to assist in this transitory period [51].

The use of ELPA for larger systems meshed well with DFT requirements - indeed, DFT was one of the motivating applications for its design. Table 3.2 lists actual runtimes for a single decomposition step with given simulation parameters, performed on Intel Knight's Landing (KNL) nodes as part of Stampede2 at the Texas Advanced Computing Center. Similar results were obtained on Cori Phase 2 at the National Energy Research Scientific Computing Center. These results, which utilized a version of ELPA (2016.05.004) which had only fledgling KNL support, demonstrate a sweeping success. A future implementation improvement would be utilizing multiple subteams of processes to solve problems too small to solve with the full list of ranks.

Algorithm 6: A projection method that tiles the resulting products of $H\mathbf{V}_{:,j}$ while communicating information for the next stage in order to utilize ?GEMM calls while also overlapping communication with the computation. Uses a single global summation reduction, which can be split into multiple calls in order to avoid out-of-memory buffer limitations in limiting cases.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$

Output: Rayleigh projection matrix $\mathbf{G}$

setup n_{buffer} initial buffers;
work $\mathbf{U} \in \mathbb{C}^{N/P \times n_{\text{buffer}}}$;
for $j = 1$ **to** s **step** n_{buffer} **do**
    $\mathbf{U} = H\mathbf{V}_{:,j:j+n_{\text{buffer}}-1}$;
    pack & start comms. for $H\mathbf{V}_{:,j+n_{\text{buffer}}:j+2n_{\text{buffer}}-1}$;
    $\mathbf{G}_{j:s,j:j+n_{\text{buffer}}-1} = \mathbf{V}^H_{:,j:s}\mathbf{U}$
end
global sum $\mathbf{G}$ over all MPI processes

3.1.3 Update

Update uses the resulting Ritz vectors $\mathbf{W}_i$ to refine the initial basis $\mathbf{V}$. While the update step is not strictly necessary, it does tend to improve convergence [45]. It does so at a considerable total cost of Ns^2 FLOPs, especially when memory limitations prohibit a second temporary buffer of $\mathbf{U}^{p_k}$. Using the ScaLAPACK machinery tends to be cost prohibitive, as the structures $\mathbf{V}$ and $\mathbf{W}$ are large, and of differing dimensioned parallel decompositions. Yet when memory allows, the choice of a decomposition only along N means that for a resulting $\mathbf{W}$ gathered to each process, there is no additional communication during the ?GEMM operation. This is ideal as discussed in prior sections. As a demonstration of this effect, see Table 3.4.

Algorithm 7: Accounting for requiring separate work tiles for a distributed $\mathbf{G}$ where the global no longer fits in memory changes Algorithm 6 drastically thanks to the requirement of global reduction per work tile. The 2D block cyclic distribution of $\mathbf{G}$ requires owners to store relevant regions of the work tile to their own local $\mathbf{G}^{\{p\}}$. Multiple G_{work} buffers could be used if Ψ was distributed across both dimensions.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$, general communicator info F
Output: Distributed Rayleigh projection matrix $\mathbf{G}$

setup n_{buffer} initial buffers;
work $\mathbf{U} \in \mathbb{C}^{N \times n_{\text{buffer}}}$;
setup $G_{\text{work}} \in \mathbb{C}^{s \times n_{\text{buffer}}}$;
for $j = 1$ **to** s **step** n_{buffer} **do**
  $\mathbf{U} = H\mathbf{V}_{:,j:j+n_{\text{buffer}}-1}$;
  pack & start comms. for $H\mathbf{V}_{:,j+n_{\text{buffer}}:j+2n_{\text{buffer}}-1}$;
  $G_{\text{work}} = \mathbf{V}^H_{:,j:s}\mathbf{U}$; `// ?GEMM`
  reduce G_{work} in place across all contributors, then broadcast to associated block cyclic columns ;
  effect $\mathbf{G}^p_{F(j:s,j:j+n_{\text{buffer}}-1,p)} = G_{\text{work}}$;
  `//` $F : (i, j, p) \to (i_d, j_d, (i_p, j_p))$ `performs the appropriate mapping 1D to 2D block cyclic`
end

Currently, there are three methods of updating $\mathbf{V}$. Algorithm 9 marks the classic way; as $\mathbf{W}$ could typically be entirely locally stored, even later when solving DCMP in parallel. UPDT actually benefits from the 1D parallel decomposition of the wave functions into $\mathbf{V}$ up to fairly large problems, as once $\mathbf{W}$ is local, the rest of the entire operation is a call to a local ?GEMM, eliminating further communication and maximizing floating point throughput. Futhermore, the temporary basis $\mathbf{U}$ is potentially still aligned and allocated,

Algorithm 8: A proposed naive projection algorithm for a $\mathbf{V}$ decomposed over both N and s. This is contingent on $H\left[\rho\left(\mathbf{r}\right)\right] : \mathbb{C}^N \rightarrow \mathbb{C}^N$. Note how the required communication within the dot products when $j \notin s^p, s^p \subset s, p \in P$ leads to a serious ordering requirement thanks to memory limitations and avoiding duplications of Ψ.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$

Output: Rayleigh projection matrix $\mathbf{G}$

$s^p \leftarrow \{k \in s | \mathbf{V}_{:k} \text{ is stored locally}\}$;
for $j = 1$ **to** s **do**
  $p_j \leftarrow \{p \in P | j \in s^p\}$;
  $\mathbf{u} = H\mathbf{V}_{:,j}$
  p_j sends $\mathbf{u}$ to $P \setminus p_j$;
  for $i = j$ **to** s **do**
    $\mathbf{G}_{ij} = \mathbf{V}_{:,i} \cdot \mathbf{u}$
  end
end
for $j = 1$ **to** s **do**
  global sum $\mathbf{G}^p_{j:s,j}$ over all MPI processes $p \in P$
end

avoiding the eventual case where fragmentation causes an allocation failure for such a large object. In the case of platforms with fast memory where $\mathbf{V}$ does not exist in fast memory, it sometimes increases speed to manually allocate the buffer in fast memory. Unfortunately, even on Intel Knights Landing where the MCDRAM provides massive incentives for this practice, the actual hardware in HPC clusters is rarely configured such that manual control is allowed in larger node allocations; for this reason, we have avoided this practice for now.

Again, the differing blockcyclic distributions of $\mathbf{V}$ and $\mathbf{W}$ cause typically used typical parallel library frameworks, like PBLAS, to require enough

Molecule	# Node	# MPI	# SMT	N [1e6]	s [1e4]	t [secs]
$Si_{3917}H_{1036}$	16	64	17	1.45	1.0	8
$Si_{3917}H_{1036}$	16	64	17	2.93	1.0	9
$Si_{3917}H_{1036}$	16	64	17	1.45	1.5	16
$Si_{3917}H_{1036}$	16	64	17	2.93	1.5	16
$Si_{10701}H_{1996}$	64	256	17	3.02	2.6	26
$Si_{20389}H_{3076}$	256	512	34	5.35	5.1	65

Table 3.2: Data describing time to solution for the decomposition step in parsec-1.4.x on Stampede2 at the Texas Advanced Computing Center. Simulations with $s < 128$ or $16\sqrt{n_{MPI}} < s$ automatically utilize a system provided flavor of lapack instead of ELPA-2016.05.004.

creative usage causing additional syncing or memory movement as to render them inefficient, as either $\mathbf{V}$ or $\mathbf{W}$ can cost the majority of memory on a node.

In cases where the execution of Algorithm 9 would not fit into memory, we can instead opt to reduce the size of the buffer $\mathbf{U}$, and tile the remaining operations as in Algorithm 10.

At increasingly large simulation sizes, $\mathbf{\Psi}$ has a 1D parallel distribution along only number of grid points instead of both number of grid points and number of states as $\mathbf{V}$ causes a fully in-memory algorithm for UPDT without replication of methods for a given distribution type, or potentially out of DRAM algorithms as in [36–38].

Algorithm 9: The classic local update with aggressive memory usage. Full duplicates of $\mathbf{V}^{p_k}$ and $\mathbf{W}$ allow for maximum arithmetic performance given pre-existing data layout through the use of a ?GEMM.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Ritz vectors, $\mathbf{W} \in \mathbb{C}^{s \times s}$

Output: Refined basis, $\mathbf{V}$

work $U \in \mathbb{C}^{N \times s}$;
$\mathbf{U} = \mathbf{V} * \mathbf{W}$;
$\mathbf{V} = \mathbf{U}$;

3.2 Results

3.2.1 Scaling on Large Silicon Nanoclusters

Silicon as a substance is a marvelous necessity for widespread transistor development [26,28,29,52–56]. Even as we reach the limiting scale of standard transistor manufacturing for silicon, silicon quantum dots show potential as a desirable qubit material [53]. Scientists consider silicon nanoclusters exciting as the potential applications range from drug delivery vehicles, imaging agents, light emitting particles, to photovoltaics [56]. Silicon clusters warrant doping and vacancy studies. As a demonstration, I have prepared two fully self consistent $Si_{10701}H_{1996}$, computed at full resolution and without symmetry. Figures 3.2 and 3.3 demonstrate a comparison between a charged and neural $Si_{10701}H_{1996}$. Table 3.3 provides a full description of the system parameters used.

Algorithm 10: Update with memory footprint reduction. The use of a non distributed ?GEMM is still more performant than a fully distributed solutions as it still completely avoids further communication.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, and the Ritz vectors, $\mathbf{W} \in \mathbb{C}^{s \times s}$

Output: Refined basis, $\mathbf{V}$

choose $B < N, k = 0$;
work $\mathbf{U} \in \mathbb{C}^{B \times s}$;
for *each group of count B rows labeled as* $\mathbf{K}$ **do**
    $\mathbf{U} = \mathbf{V}_{\mathbf{K},:} * \mathbf{W}$;
    $\mathbf{V}_{\mathbf{K},:} = \mathbf{U}$;
end

3.2.2 Performance on Insulin

Artificial insulin is a complex biomolecule that a growing number people depend on, owing in part to increased incidence rate of diabetes in the western world. In general, biomolecules are of vast interest to the future of materials design; however, they are not only very large, typically measuring in kilodaltons of mass, but are complicated to treat with the quantum accuracy offered through DFT. LDA in particular has trouble with allowing the capture of enough charge around the charge carrying elements, which leads to very slow convergence and the sometimes emergence of charge sloshing. A generalized gradient approximation, or GGA, in place of LDA, can sometimes provide benefits in these scenarios, but nothing is guaranteed a priori. This is mostly thanks to the helpful error cancelation in LDA, where it often overestimates exchange and underestimates correlation, and means that attempts

Algorithm 11: A potential UPDT algorithm when $\mathbf{W}$ cannot be gathered locally in its entirety, instead remaining in the 2D block-cyclic distribution and requiring communication scheduling prior to arithmetic operations. This algorithm was not implemented due to time constraints, preferring Algorithm 12. This should even work in the extreme case where the full sized working unit $\mathbf{U}$ cannot be storedfurther worsens the situation.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N \times s}$, the distributed Ritz vectors, with the local window of Ritz vectors, $\mathbf{W}^{\text{loc}} \in \mathbb{C}^{s_a^{\text{loc}} \times s_b^{\text{loc}}}$

Output: Refined basis, $\mathbf{V}$

choose $B_v < N, B_w < s$;
work $\mathbf{U} \in \mathbb{C}^{B_v \times s}$;
work tiles $\mathbf{W_J} \in \mathbb{C}^{s \times B_w}$; `// ` B_w `is the blockcyclic blocking amount`
for *each group of count* B_v *rows labeled as* $\mathbf{K}$ **do**
    $\gamma = 0.0$
    for *each group of columns* $\mathbf{J}$ **do**
        gather $\mathbf{W_J}$;
        $\mathbf{U} = \gamma\mathbf{U} + \mathbf{V}_{\mathbf{K},:} * \mathbf{W_J}$;
        $\gamma = 1.0$
    end
    $\mathbf{V}_{\mathbf{K},:} = \mathbf{U}$;
end

to improve upon LDA can often display forms of error that were missing from LDA. New exchange-correlation functionals are an area of active research.

Insulin has two commonly found forms the body uses – a monomer and a hexamer, pictured in Figure 3.6. I present verifications for improvements and algorithms made earlier in this chapter utilizing insulin in its hexamer form. These verifications include the use of buffers to substantially improve timings of the Rayleigh Ritz projection algorithm in Figures 3.7 and 3.8. These results

Algorithm 12: A potential UPDT algorithm when $\mathbf{W}$ cannot be gathered locally in its entirety, instead remaining in the 2D block cyclic distribution and requiring communication scheduling prior to arithmetic operations. This fakes a 1D block cyclic distribution in V, by offsetting the last process P-1 to be the first in the descriptor.

Input: Orthonormal basis, $\mathbf{V} \in \mathbb{C}^{N_g \times s}$, the distributed Ritz vectors, with the local window of Ritz vectors, $\mathbf{W}^{\text{loc}} \in \mathbb{C}^{s_a^{\text{loc}} \times s_b^{\text{loc}}}$, temporary storage $\mathbf{U} \in \mathbb{C}^{B_v \times s}$

Output: Refined basis, $\mathbf{V}$

choose $B_v \leq N_g, B_w < s$;
$F : \langle (i_p, j_p), (n_{ip}, n_{jp}), \dim \mathbf{W} \rangle \rightarrow$
$\langle (p+1, 1), (\text{p}roc.column), (range(V)) \rangle$; `// create a ScaLAPACK descriptor that maps a 2D block cyclic grid to a faked 1D block cyclic grid`
for *each virtual column in 1D* **do**
  for *each group of count* B_v *rows labeled as* $\mathbf{K}$ *in virt.column* **do**
    $\mathbf{U} = \mathbf{V}_{\mathbf{K},:} * \mathbf{W}$; `// PDGEMM`
    $\mathbf{V}_{\mathbf{K},:} = \mathbf{U}$;
  end
end

are impressive as the larger the count of buffers, the more wasted operations were done to avoid multiple DGEMM calls. To further elucidate – there are $n_{buffer}^2/2 - n_{buffer}$ entries of $\mathbf{G}$ that are calculated twice, thanks to $\mathbf{G} = \mathbf{G}^H$ arising from the Hermitian Hamiltonian. Note that while we do not uncover any new behavior in the insulin hexamer, this tester for new routines should be considered an example proof of concept for future self-consistent work.

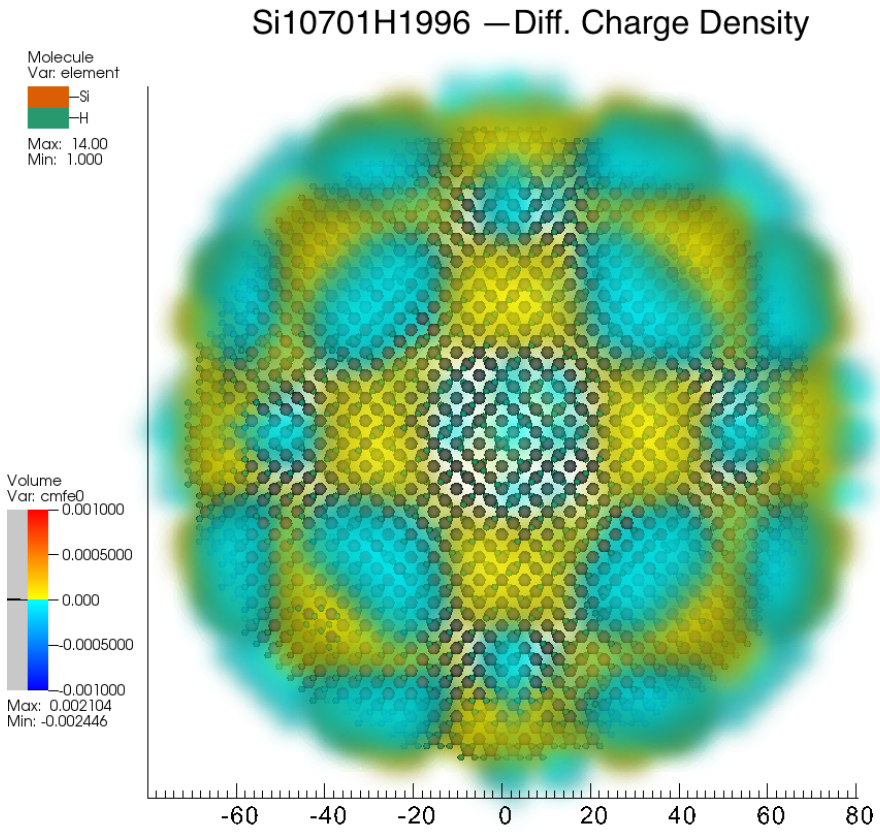

Figure 3.1: Difference in $\rho(r)$ between $[Si_{10701}H_{1996}]^{-1}$ and $[Si_{10701}H_{1996}]^{+0}$. See Table 3.3 for specifics on the two systems.

3.3 Conclusions

We explain the improvements made to the Rayleigh-Ritz process in order to produce eigenpairs at scale for parsec-1.4.x, parsec-2.0, and spdft. These improvements and full reworkings of PROJ and UPDT include the discussion

Si10K	Z=+0e	Z=-1e
s	26,000	26,000
h	0.7 a.u.	0.7 a.u.
r_Ω	80 a.u.	100.7 a.u.
N [xyz]	6.37E+6	1.267E+7
N_p	26672	14480
—{P}—	64 KNL	256 KNL
FILT [s]	1597.41	1153.56
ORTH [s]	3122.15	1381.50
PROJ [s]	516.37	223.58
DCMP [s]	45.59	35.77
UPDT [s]	75.37	36.84

Table 3.3: Charged and uncharged $Si_{10701}H_{1996}$ run on Intel KNLs. The charged silicon required a larger boundary sphere radius in order to achieve convergence, which required a larger set of KNLs to run on.

and considerations of algorithms enabling the Rayleigh-Ritz process operations at scale, as well as differences between them. Important is the explanation of why certain operations are much worse at scale given the discussed parallel decomposition along the grid points N of $\mathbf{\Psi}$, and what is done in the short term to rectify those shortcomings.

We present timing statistics for various sized silicon nanoclusters which have their surfaces passivated with hydrogen, with FLTR and ORTH timings on up to 50,000 silicon atoms, and a fully self consistent charged $Si_{10701}H_{1996}$. Manual integration of the charge density over the volume between the two

Molecule	N [1e6]	s [1e4]	FLTR	ORTH	PROJ	DCMP	UPDT
$Si_{10701}H_{1996}$	3.02	2.6	1276	1333	887	26.45	1018
$Si_{10701}H_{1996}$	3.02	2.6	1326	1182	130	43	33
$Si_{20389}H_{3076}$	5.35	5.1	3732	2512	1543	65.32	4458
$Si_{50867}H_{5748}$	12.24	12.6	1.63e4	6.7e4	N/A	N/A	N/A

Table 3.4: Early distributed timing results in seconds of the Chebyshev filtering subspace iteration's various stages for larger silicon nanoclusters. We used 64 and 256 nodes for $Si_{10701}H_{1996}$ and $Si_{20389}H_{3076}+$ respectively. The nodes were Intel KNL nodes at the Texas Advanced Computing Center, using quad cache mode and either a 4x17 or 2x34 hybrid MPI+OpenMP configuration. These results demonstrate gains when using additional memory space. Tuning for the hardware makes a huge difference, as demonstrated through the two $Si_{10701}H_{1996}$ clusters.

charge different species as $\sum_{r_i \in B_0(r)}(\rho^{[} - 1e] - \rho^{[} + 0])(r_i)\Delta V$ confirms the difference of one electron, as well as most of the differences in charge density occurring near the surface of the silicon nanocluster. Furthermore, special care was taken in visualizing the charge density in 3D space.

Finally, for the purposes of testing the algorithms, we present several potential future science applications of real space Kohn-Sham DFT as a tunable test cases for timed routines. Most notably, we demonstrate a full insulin hexamer, with more than 102 million grid points and 6900 states. The resulting timing results have large differences in performance based on adjusting the number of buffers for storing computed Hamiltonian operator results, and provides insight into future large scale calculations.

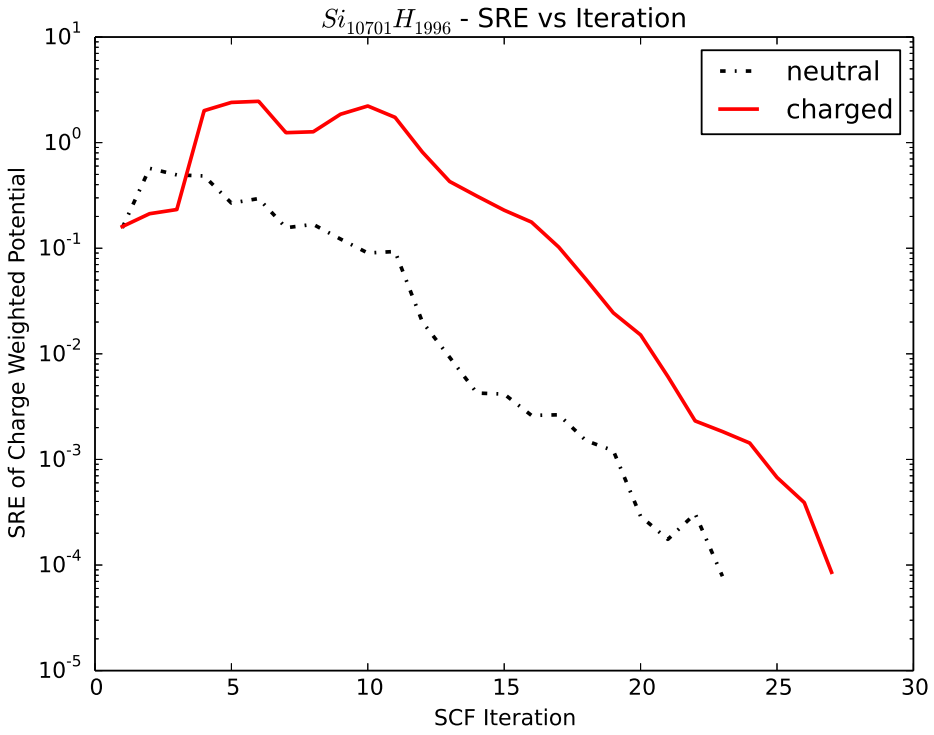

Figure 3.2: Progression of the squared residual error vs iteration for $[Si_{10701}H_{1996}]^{+0}$ and $[Si_{10701}H_{1996}]^{-1}$. See Table 3.3 for specifics on the two systems.

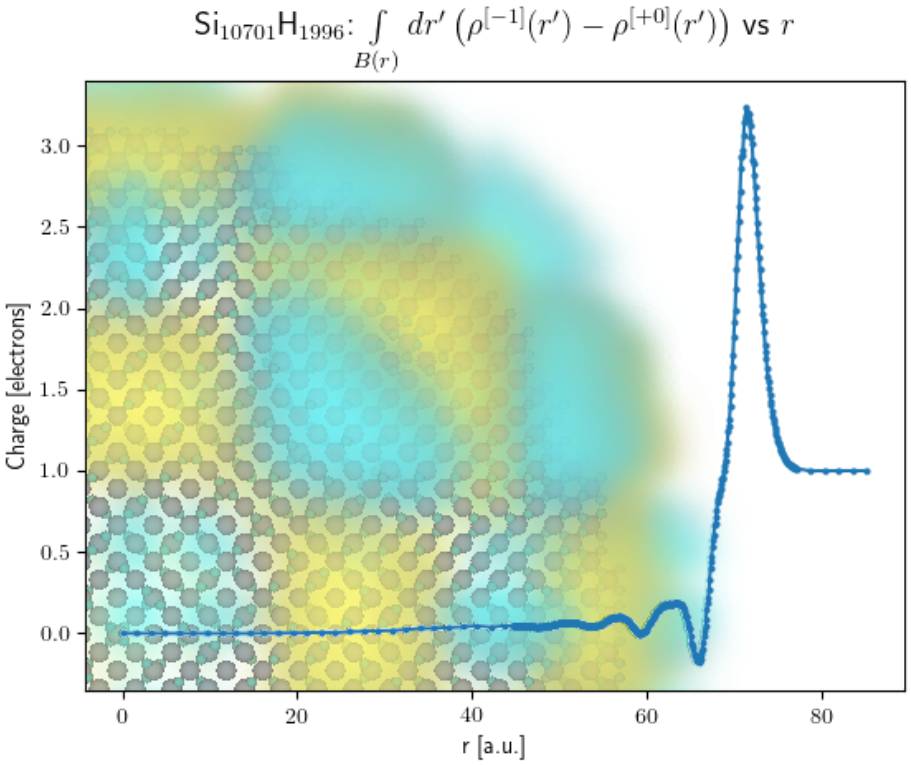

Figure 3.3: The difference in cumulative charge over a ball of radius r vs the radius r for $[\mathrm{Si}_{10701}\mathrm{H}_{1996}]^{+0}$ and $[\mathrm{Si}_{10701}\mathrm{H}_{1996}]^{-1}$. We can clearly see that the added charge has an impact near the surface. See Table 3.3 for specifics on the two systems.

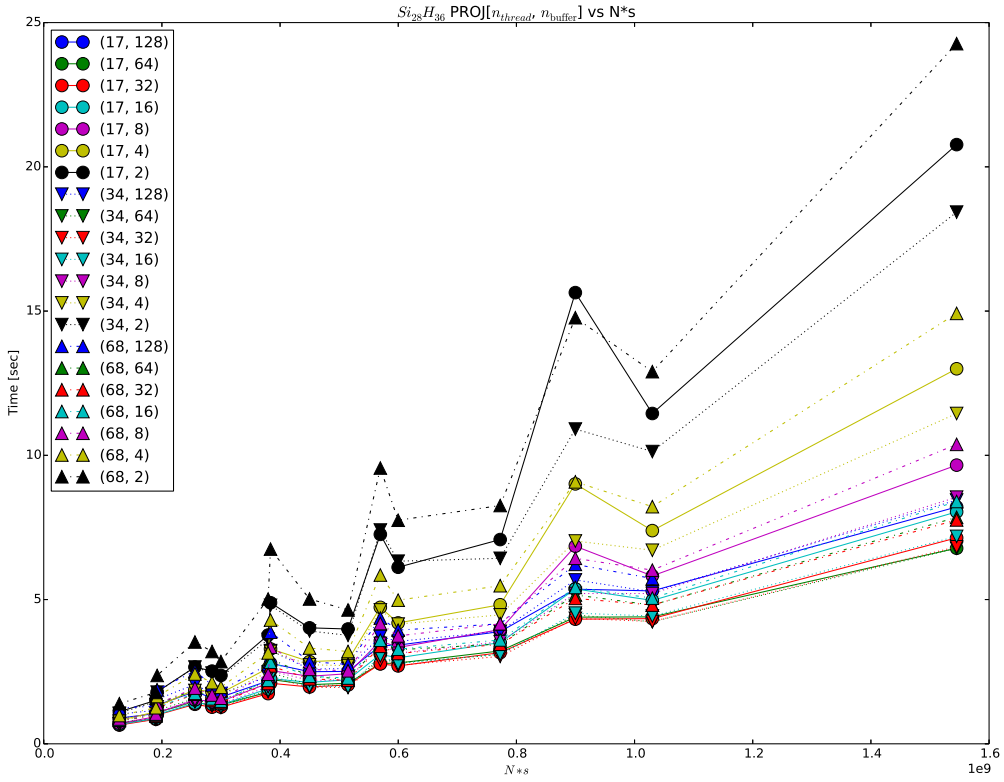

Figure 3.4: Parameter sweep exploring the effect of buffer at smaller scale through small $Si_{28}H_{36}$ at h=.25. Each line depicted is parameterized by the (no.threads, no.buffers). At this scale, 34 Hardware Threads / MPI Rank (HT/core=2) on KNL is notably better than ether 17 (HT/core=1) or 68 (HT/core=4). Notice that correct choice here can cause PROJ to have an 80% reduction in runtime.

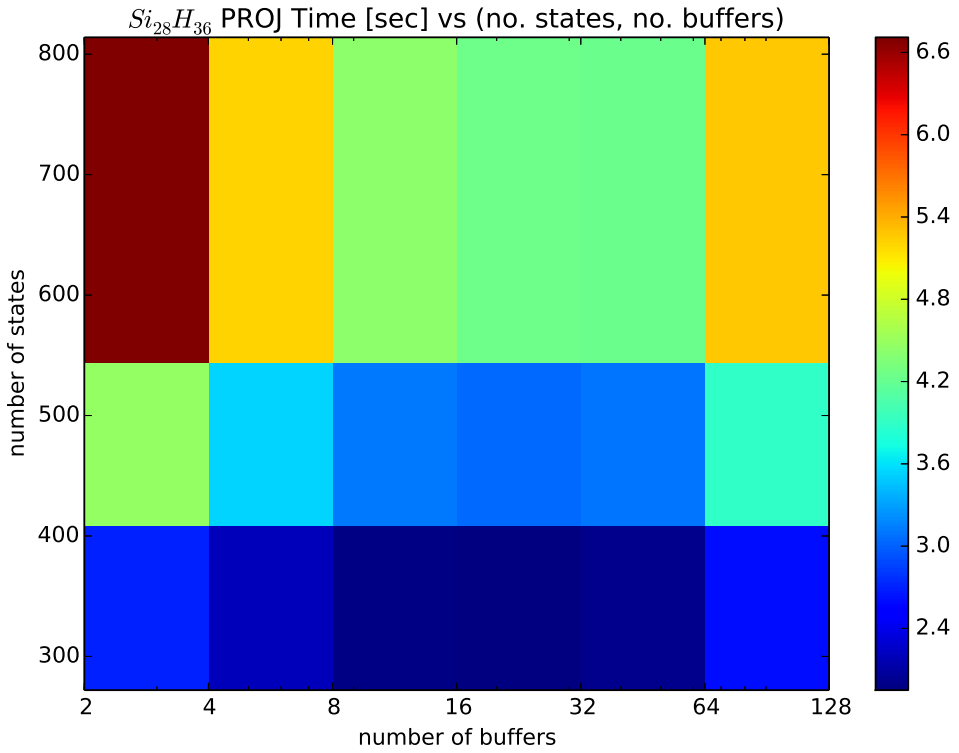

Figure 3.5: Parameter sweep exploring the effect of buffer at smaller scale through small $Si_{28}H_{36}$ at h=.25. Notice that correct choice here can cause PROJ to have an 30% reduction in runtime. While not as significant effect as on larger systems, like insulin, we still consider this a massive savings for anything larger than trivial test systems.

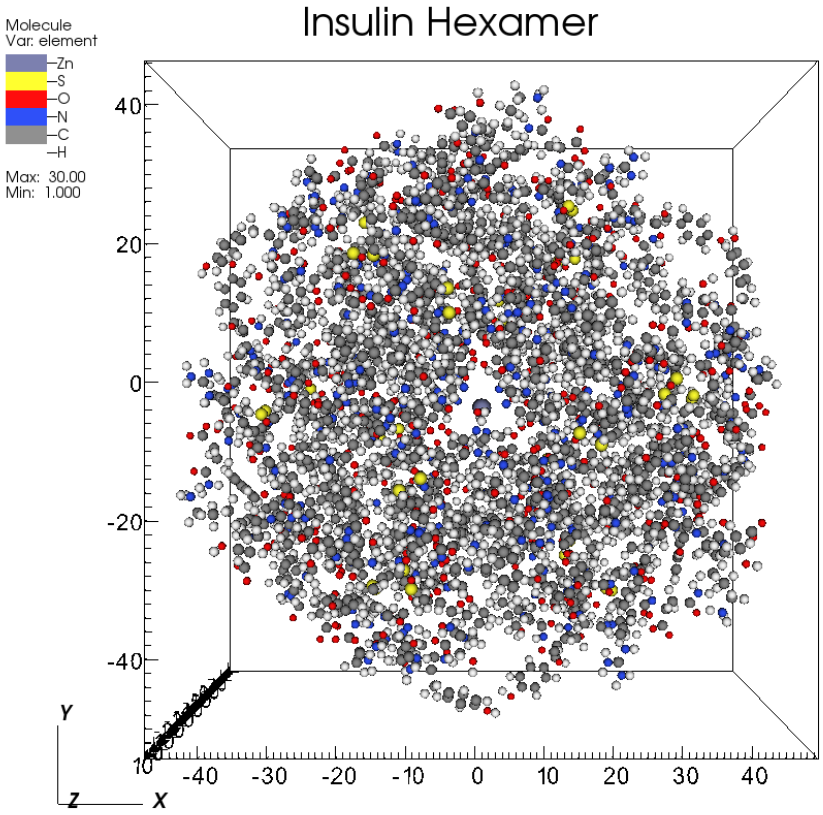

Figure 3.6: Visualization of the insulin hexamer used for large scale testing. System had between $N = [1.2e8, 1.02e8]$ grid points, and $s = [6901, 7000]$ states.

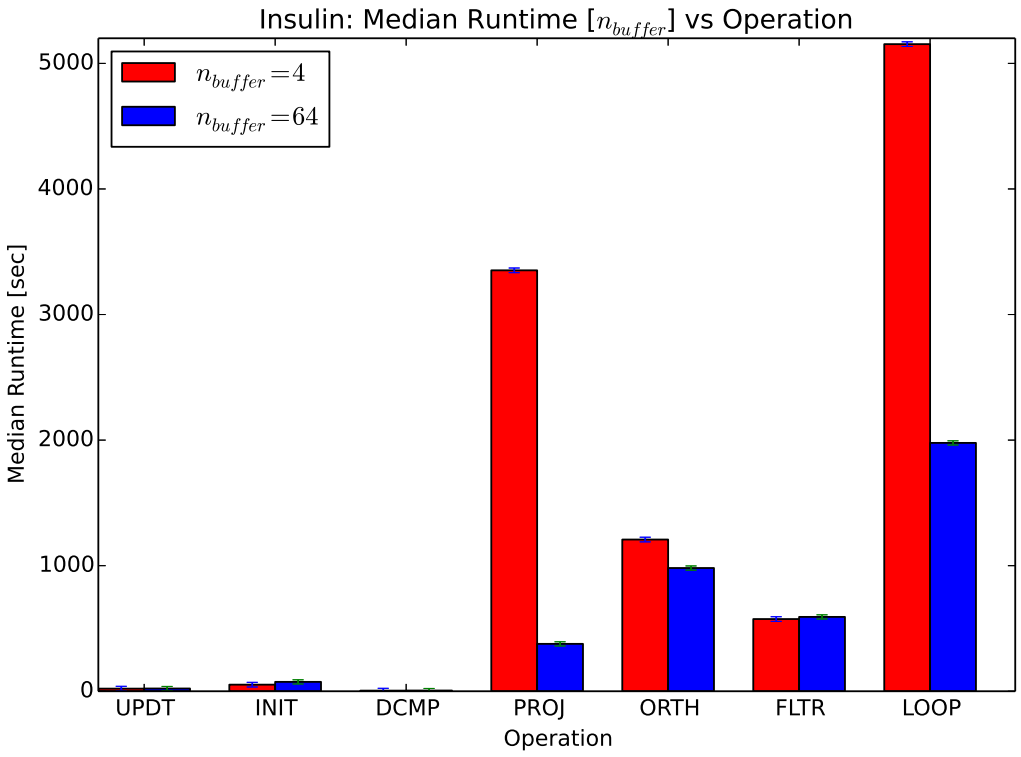

Figure 3.7: Testing PROJ on an insulin system with Stampede2, 256 KNLs in a 2 MPI ranks x 34 OMP Threads configuration, 104.3 million grid points, and 6901 states. Three filtering iterations were averaged in this test for timings. The difference in runtime for PROJ with buffer choice of 64 compared to 4 is stark.

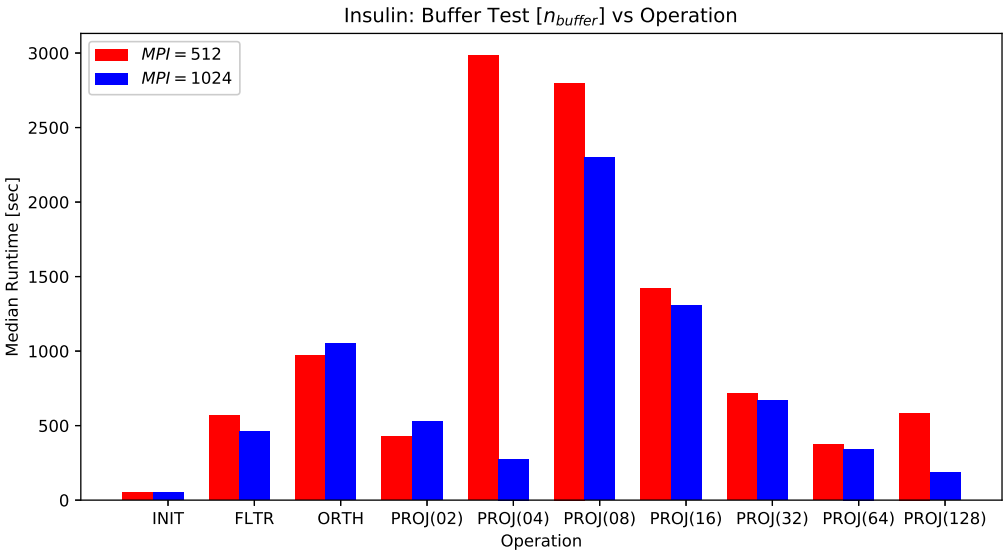

Figure 3.8: Testing PROJ on an insulin system with Stampede2, 256 KNLs in a 2 MPI ranks x 34 OMP Threads or in a 4 MPI ranks x 17 OMP Threads configuration. Still about 104.3 million grid points, and 6901 states. This test ran the typical INIT-FLTR-ORTH, and then deviated from a single PROJ run into a variety of statically coded variants that were called one after the other. The difference in runtime for PROJ as a function of buffer is stark, though the results make clear that refactoring and designing for as much cache coherence, and therefore matrix operations, as possible.

Chapter 4

Higher Order Stencils for Accurate Force Calculations

One of the major material characteristics that DFT provides are accurate interatomic forces. Unfortunately, while DFT can see the energy of a system converge relatively rapidly, the derived interatomic forces converge much more slowly, and exhibit much more sensitivity to the problem parameters. Demand for accurate forces in various material properties calculations, such as those that provide structural relaxation, vibrational modes, spectroscopic simulations, and other approaches to materials' design, are cause for more expensive DFT calculations. In real space, this manifests as a finer grid spacing h; as previously discussed, the number of grid points increases proportionally as $N = O(h^{-3})$. This drives a substantial increased computational cost for both memory and required FLOPs.

The first part of this chapter contains, in large part, a reproduction of work published as Bobbitt, N. S., Schofield, G., Lena, C., and Chelikowsky, J. R. (2015). High order forces and nonlocal operators in a Kohn-Sham Hamiltonian. Physical Chemistry Chemical Physics. [22]. Author Contributions: Bobbitt - performed excited states calculations and wrote manuscript, Schofield -

theoretical solutions to the higher order stencil formulations, main implementor of code, Lena - theoretical solutions to the higher order stencil formulations, performed ground state calculations and convergence trials, and Chelikowsky supervised the project.

The second part of this chapter addressing molecular dynamics contains work in preparation for submission as Neitzel, J., Lena, C., Schofield, G., and Chelikowsky, J. R. (2017). Improved molecular dynamics calculations using higher-order forces in real space density functional theory. *In preparation*. Contributions: Neitzel - performed molecular dynamics calculations, manuscript writing, Lena - performed molecular dynamics calculations, manuscript writing, further code development, Schofield - supervisory consulting role, and Chelikowsky - supervised the project.

4.1 Introduction

In general, interatomic forces are related to the derivative of the energy of a system with respect to the labeled coordinates: $F_J = \frac{\partial E}{\partial \mathbf{R}_J}$. We have a few choices of methodology when calculating interatomic forces in DFT. We can calculate interatomic forces with finite difference quotients of the total energy. Doing so requires a large amount of additional calculations; more specifically, for an M atom system without symmetry and a basic central finite difference approach, we would require $6M + 1$ converged DFT calculations, as depicted on a per axis basis in Equation 4.1.

$$F_{J,m} \approx \frac{E[R_J = R_J - h_m] - 2 * E[R_J] + E[R_J = R_J + h_m]}{2h_m} \tag{4.1}$$

Where h_m is the displacement on a given m-th coordinate axis. Obviously, this FDM approach is unacceptable within a molecular dynamics framework requiring DFT accuracy due to incurring a $6M + 1$ prefactor on DFT's $O(N^3)$ computational cost.

Preferably, we can calculate forces far more efficiently as a derivative of the Kohn-Sham total energy (4.2), and then the respective Hamiltonian contributions(4.3) utilizing the Hellmann-Feynman theorem [57]. This theorem allows us to disregard the dependencies of the wave function and density, $\psi_i(\mathbf{r})$ and $\rho(\mathbf{r})$ respectively, on the nuclear coordinate R_J. The remaining derivative applies only to the ionic potential, which in pseudopotential DFT results from the local and nonlocal contributions of the pseudopotential $V_{ion} = V_{loc} + V_{non-loc}$. The reader should note the existing limitations when applying this theorem, notably the requirement of full convergence of the wave functions for its validity.

$$E_{KS} = \sum_{i}^{occ} E_i - \frac{1}{2}\int \rho(\mathbf{r}) V_H(\mathbf{r}) d^3r - \int \rho(\mathbf{r}) V_{xc}[\rho(\mathbf{r})] d^3r + \int \rho(\mathbf{r}) \mathcal{E}_{xc}[\rho(\mathbf{r})] d^3r \tag{4.2}$$

where V_{ion} represents the ionic pseudopotential, V_H is the Hartree potential, and V_{xc} is the exchange-correlation potential. Taking the derivative of the energy, the force on the J^{th} atom is

$$F_J = -\nabla_J \sum_i \langle \psi_i(R_J) | H(R_J) | \psi_i(R_J) \rangle, \tag{4.3}$$

where R_J is the nuclear coordinate and H is the Hamiltonian, defined below.

$$H(R_J) = -\frac{1}{2m}\nabla^2 + V_{\text{ion}}(R_J) + V_H(\rho(R_J)) + V_{xc}(\rho(R_J)) \tag{4.4}$$

The Hellmann-Feynman theorem [57] allows us to disregard the dependencies of the wave function and the density on the nuclear coordinate R_J. The remaining derivative applies only to the ionic potential, V_{ion}. We employ Improved Troullier-Martins pseudopotentials, with the implemented form below, and multiple atoms M provide independent contributions to V_{ion} of the same form.

$$V_{\text{ion}} = \begin{cases} V_{\text{loc}}(r) + \sum_{l,m} \dfrac{|V_l'(r)\varphi_l(r)Y_{l,m}(\theta,\phi)\rangle\langle Y_{l,m}(\theta,\phi)\varphi_l(r)V_l'(r)|}{\langle \varphi_l(r)|V_l'(r)|\varphi_l(r)\rangle} & r \le r_c \\ \dfrac{-Z}{r} & r > r_c \end{cases}. \tag{4.5}$$

The kernel in the projection operation in Equation 4.5, $\langle Y_{l,m}(\theta,\phi)\varphi_l(r)V_l'(r)|$, is zero outside the $B_J(r_c^l)$, ball of radius r_c^l centered on the atomic position $\mathbf{R}_J$, where r_c^l is the angular momentum associated cutoff parameter. Typically, r_c^l is between 1.0 – 3.0 a.u. for most common elements. Therefore, integration of the nonlocal component is only required within the sphere enclosed by this cutoff radius. This integration has been approximated with a Riemann sum,

and then a single point Gauss quadrature applied for any contributing element of volume h^3; we refer to this as the low order method.

$$\int_{\Omega} f(r)\,\psi(r)\,d^3r = \sum_{r_{xyz}\in\Omega} f(r_{xyz})\,\psi(r_{xyz})\,h^3 + E(h) \tag{4.6}$$

In the higher order stencil method, we treat each volume integration piece of the Riemann sum with higher order Gauss quadrature points. We do not wish to calculate these points, nor do we wish to maintain any record of them, as memory considerations render such actions prohibitively expensive. A series of virtual points are introduced, and the value of $\sum w_i\psi(r+x_i)$ is approximated through the use of Taylor series expansions, built *in situ*, with the higher order derivatives in the Taylor series expansion solved through information from neighboring values of ψ on the actual grid. We express the overall sum of these outer products as the contribution $z_J z_J^T$ for a vector z_J of nonlocal contributions from nuclei J. This method allows us to simulate a finer grid without incurring the memory penalty or overall computational expense of orthonormalizing all of the extra points, and delays reaching the cubic scaling behavior.

We employ the improved Troullier-Martins pseudopotentials, [10] which take the form in Equation (1.14). The integration around the single calculated wave function data point transforms from a simple 1 point Gaussian formula to a multiple point Gaussian formula that uses wave function values expressed in terms of the Taylor series expansions. As explored in our earlier paper, [22] this

allows us to utilize more of the much more densely available pseudopotentials, resulting in a tunable calculation that either converges to a more accurate solution in fewer SCF iterations at similar grid spacings, or converges to an accurate solution at higher grid spacings. We calculated carbon monoxide using both lower order and higher order approaches within DFT. The results (Figure 4.9) showcase the potential of the higher order stencils treatment.

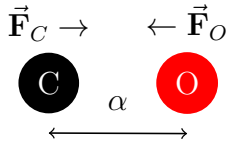

Figure 4.1: An example CO molecule with bond length α. In improving our force stencils, this was our most common testing molecule.

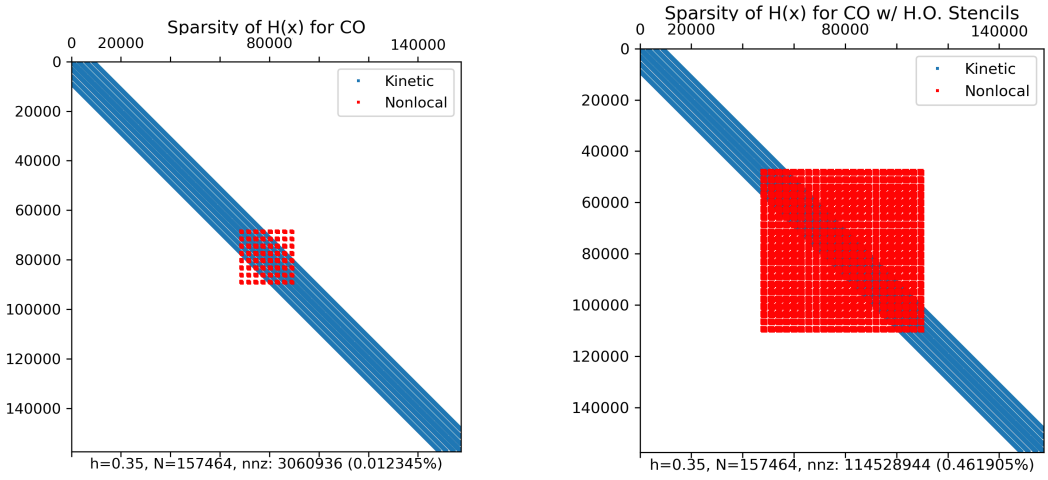

Figure 4.2: Sparsity plot for CO depicting the additional computational intensity generated from the higher order stencils. There is also a notable amount of fill in if storing the full outer products zz^T resulting from the $v_{nonlocal}$, which is typically avoided through storing the vector z instead. The vector z should be the length of the measure of points in $B_J(r_c^l)$.

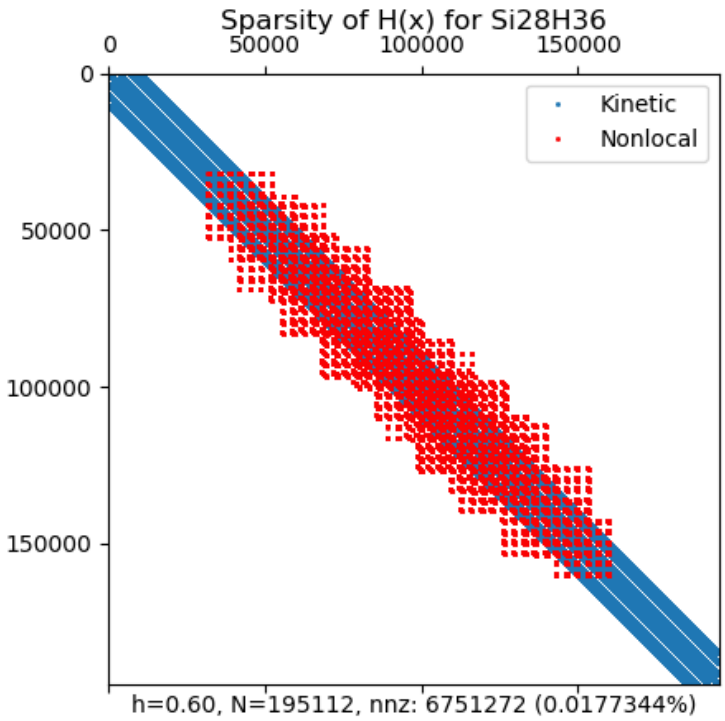

Figure 4.3: Sparsity plot for the Hamiltonian for a $Si_{28}H_{36}$, depicting the laplacian contributions and nonlocal contributions separately. Other contributions to H influence only the diagonal.

4.2 Molecular Dynamics

Molecular dynamics (MD) serves as a popular computational technique for the study of a variety of systems including DNA [58], quantum dots [59], liquid crystals [60], cavitation bubbles [61], and aromatic hydrocarbons. [62] Resulting material property calculations tend to be a function of MD simulations with thousands of iterations; within *ab initio* density function theory (DFT), each MD iteration consists of a fully self-consistent loop that can cost

large amounts of computer time. We previously have published papers on using Chebyshev filtering to accelerate the convergence of real space density functional theory with pseudopotentials. [63, 64]

Applying this to the molecular dynamics framework is a straightforward process. We integrate the atomic motion through a standard velocity verlet algorithm, [65] at velocities given by a standard Boltzmann distribution.

Another issue with MD calculations common to any numerical integration method is the accumulation of small errors over longer time periods. These errors cause the total energy of a system to drift. [66, 67] While integration of floating point error cannot be avoided over time, accumulation of method error should be avoided. As such, the energy drift over time is a common measure of the stability and level of convergence of an MD simulation. Achieving this higher level of convergence in density functional theory requires either a larger energy cutoff when using plane waves or a finer grid spacing (h) in the case of real space calculations. A finer grid spacing ($h_{fine} < h_{coarse} < 1$) is expensive because the computational cost scales roughly as the cube of the inverse grid spacing, $O(h^{-3})$.

We recently published an improved method of calculating forces [22], based on high order integration techniques, which allows the computation of more accurate forces while using fewer grid points. This development is important for MD applications as the reduction in error in the force calculations reduces both the energy drift over time as well as center of mass drift. Furthermore, with our new high order technique, this improvement in accuracy can be

obtained with a coarser grid spacing, which reduces the real world cost of each step of the MD simulation, with either fewer SCF iterations to a converged MD iteration or cheaper SCF iterations on the whole.

4.3 Results

For all tested molecules and parameters the high order integration scheme (HO) conserved energy better than the traditional low order method (LO). Energy conservation was tested by examining total energy evolution for the O_2 and C_6H_6 molecules across different values of the grid spacing parameter (h). In each case we excite vibrational modes of the molecule, impart no initial kinetic energy, then integrate forward in time with timesteps of 1.0 fs.

The simplest such test was for an O_2 molecule with an excited vibrational mode along the direction of the bond. A few example trajectories of such excitations are plotted in Figure 4.10 and show both the desired energy evolution, the periodic bounded evolution observed for $h = 0.3$, and an undesirable and unreliable evolution, the irregular unbounded case for LO with $h = 0.4$. These evolutions are also the first piece of evidence for the relative strength of the HO scheme, as the irregular evolution only occurred in any of our test cases for the LO algorithm.

For every grid spacing value where regular oscillations were measured, the fractional size of the oscillations was smaller for the HO implementation. In fact, the HO scheme was able to beat the 10% maximum drift heuristic with $h = 0.3$ where the LO scheme requires $h = 0.2$ to beat the same metric.

Performing a similar analysis for benzene, C_6H_6, demonstrates the strength of the HO scheme for larger and more complex molecules. For these molecular dynamics computations we intentionally excited a limited number of longitudinal modes of the molecule to simplify normal mode analysis. Energy drift over time is shown in Figure 4.11 with $h = 0.3$, the value for which we see the most benefit from the HO integration. This case shows substantial reductions in the magnitude of energy drift for the HO approach compared to the LO.

4.4 Conclusions

We examine the effect of higher order stencils on several improvements to Kohn-Sham DFT, including pseudopotential contributions through the use of derived higher order stencils. In particular, the improvements made to pseudopotential Kohn-Sham DFT in real space allow for great gains of accuracy at the expense of additional computational requirements. Recognizing that for large counts of M in certain medium simulation scale situations, the Hamiltonian operator scales as $O(N + M^2)$, we see that the number of grid points surrounding the nuclei greatly impacts the computational and communication costs of the Hamiltonian operator.

We explore tuning the prefactor of the M with changes that allow for the decrease in a cutoff radii r_c^l. This result has large impact for load balancing concerns at exascale. Importantly, the increased computational requirements are well understood within the concept of a stored sparse Hamiltonian, which

we visualize for the reader. Through the sparsity graphs, we compare the higher order stencils with varying r_c^l of the pseudopotential to formulating a multistep application of the Hamiltonian in the form of $H^2(x)$. We have furthered an important understanding of the higher order stencils with regards to the increased amount of data required, informing internode parallelism design decisions going forward regarding both algorithm design of pseudopotential real space DFT and design of future pseudopotentials.

We apply higher order stencils to both energy minimization and full molecular dynamics runs. The higher order stencil reformulation of the nonlocal component of the ionic potential improves on the level of grid spacing for total energy convergence. The amount of energy drift in tested runs decreased with the use of higher order stencils compared to their low order in the single molecule test cases, O_2 and C_6H_6. The higher order stencil's reduction of energy drift within a molecular dynamics framework demonstrates the efficacy of such higher order methods when requiring accurate, consistent forces. These results suggest that interested parties should examine any parallel performant formulation of higher order stencils for use in even larger simulations on new systems where memory per core is at a premium.

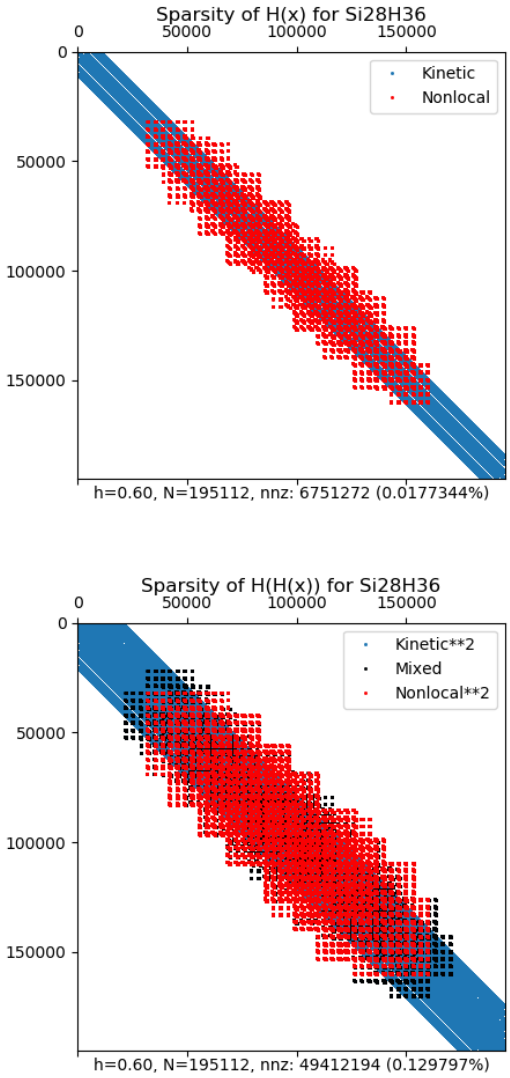

Figure 4.4: Real space nonlocal calculation grid point contributions for $H(H(x))$ instead of $H(x)$. Demonstrates required data to attempt to encapsulate two applications of the Hamiltonian operator in a single operator, without communication between those two applications. The large increase in the amount of interaction as shown by the increase in the bandwidth of the Hamiltonian renders this strategy difficult.

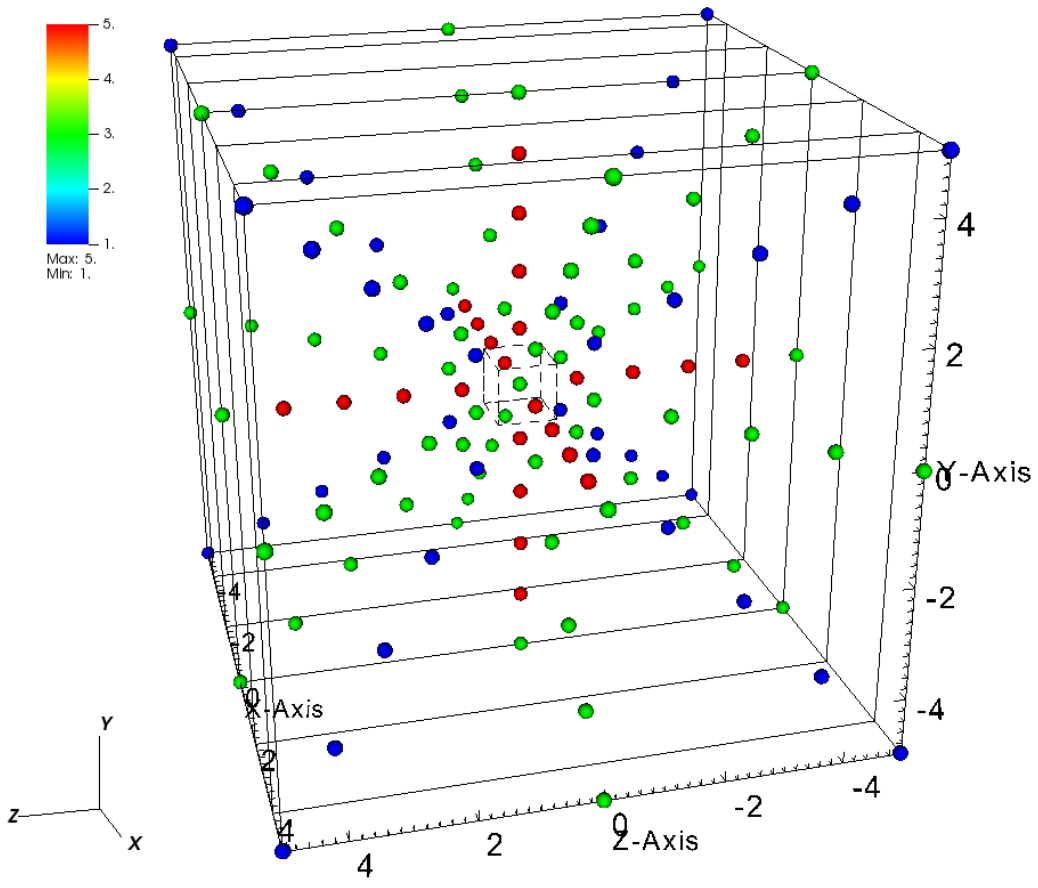

Figure 4.5: Real space finite difference operator stencil grid point contributions for a single point's contributions to the nonlocal integral. Colors depict number of times a points is referenced in the sum of the stencils.

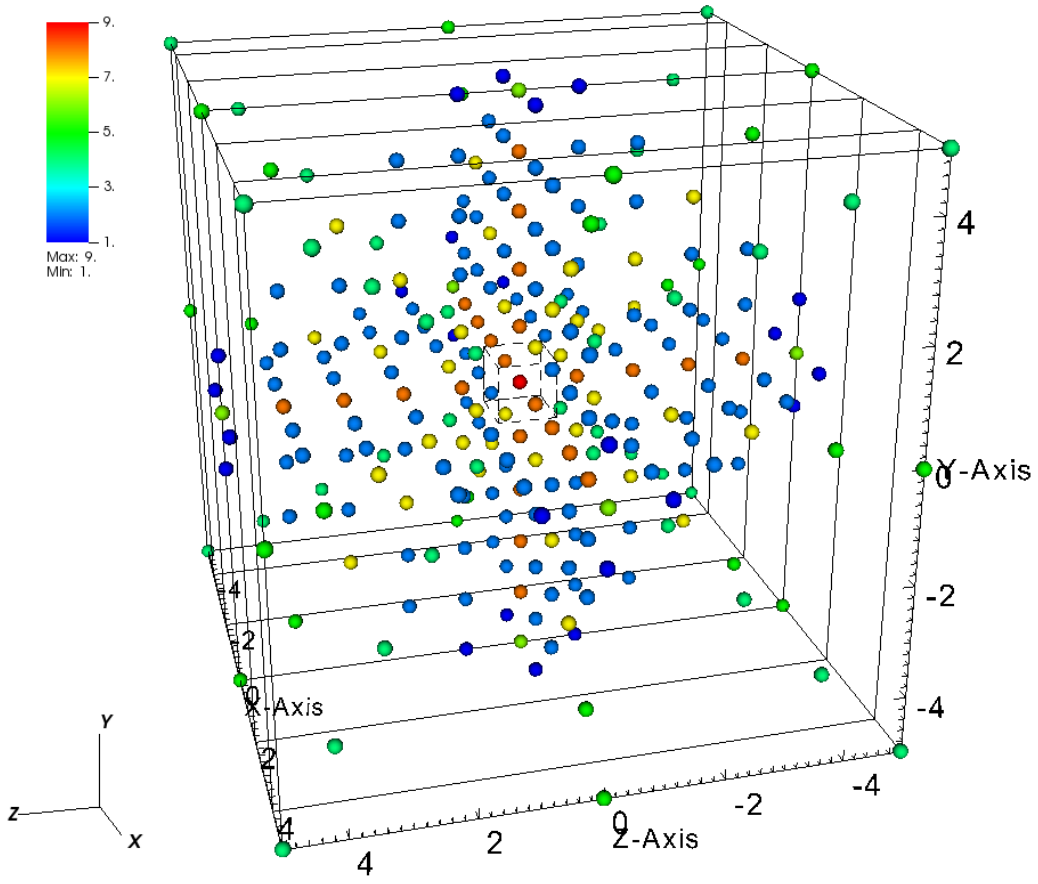

Figure 4.6: Real space finite difference operator calculation grid point contributions for one iterations worth of points. Colors depict number of times a points is referenced in the sum of the stencils.

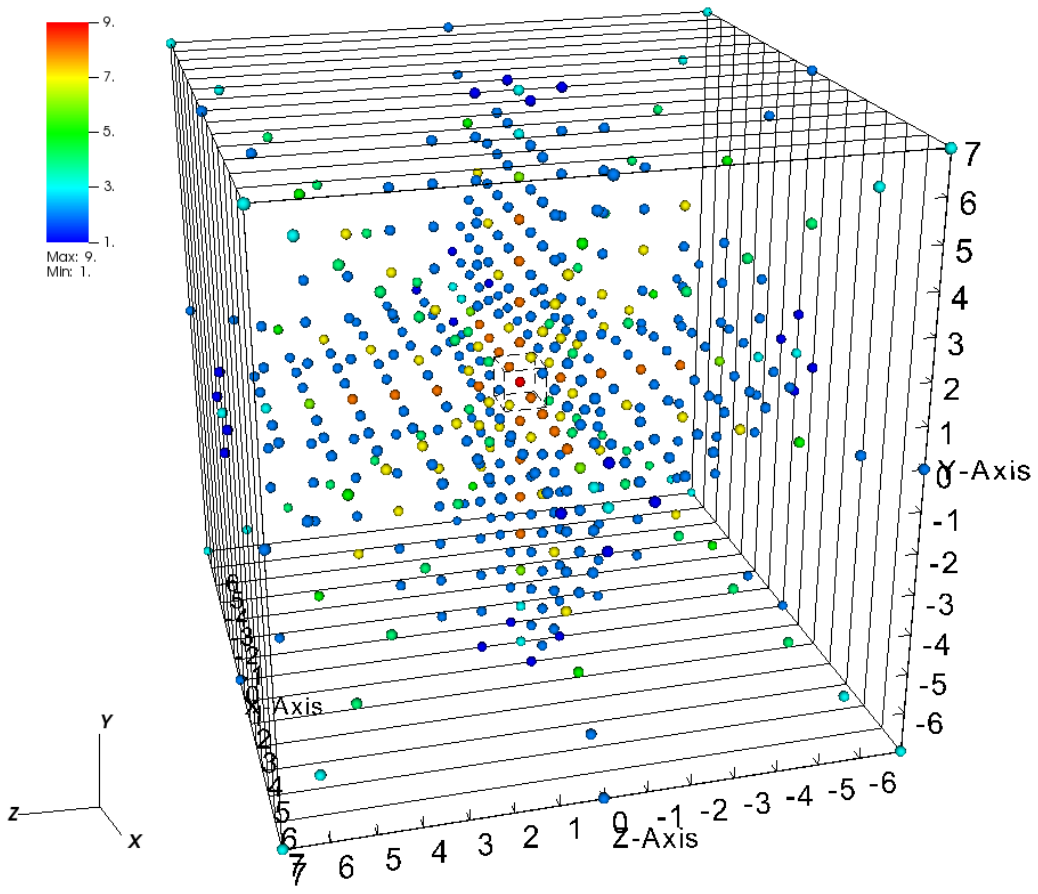

Figure 4.7: Real space finite difference operator calculation grid point contributions for one iterations worth of points. Colors depict number of times a points is referenced in the sum of the stencils.

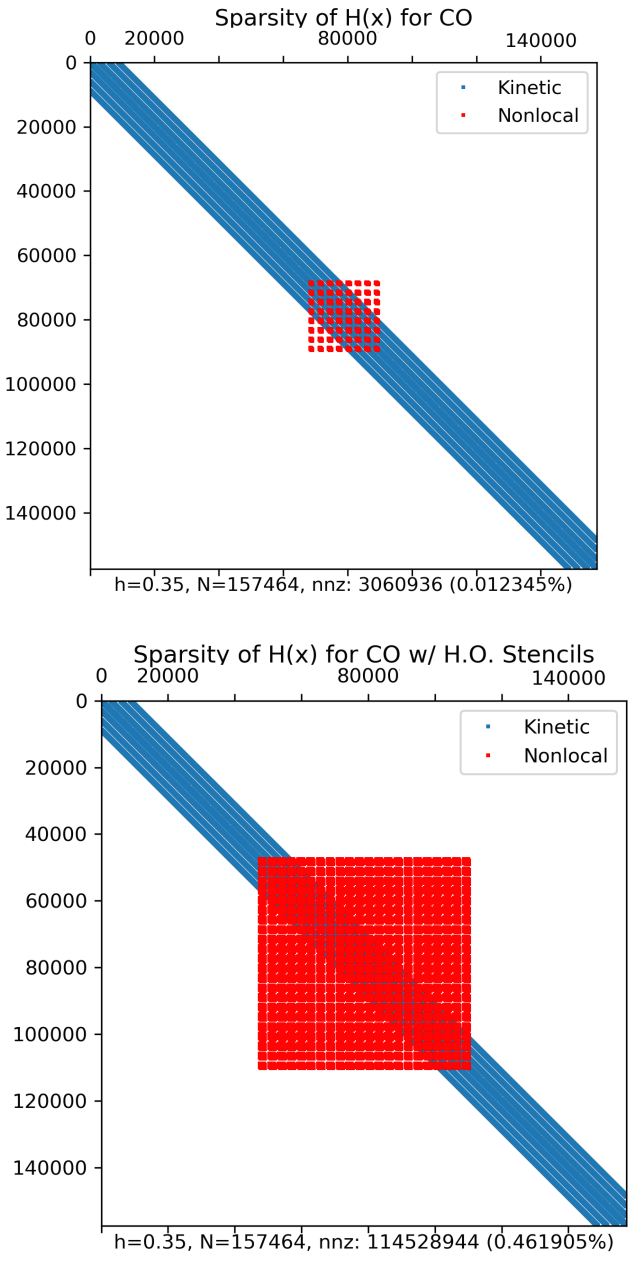

Figure 4.8: Sparsity plot for carbon monoxide. The higher order stencils on a virtual grid fill out the Hamiltonian through references to actual grid points, causing the number of nonzero entries to increase dramatically. As the new entries result from an outer product zz^T, to save memory it is preferable to store only the vector z and calculate the products on demand.

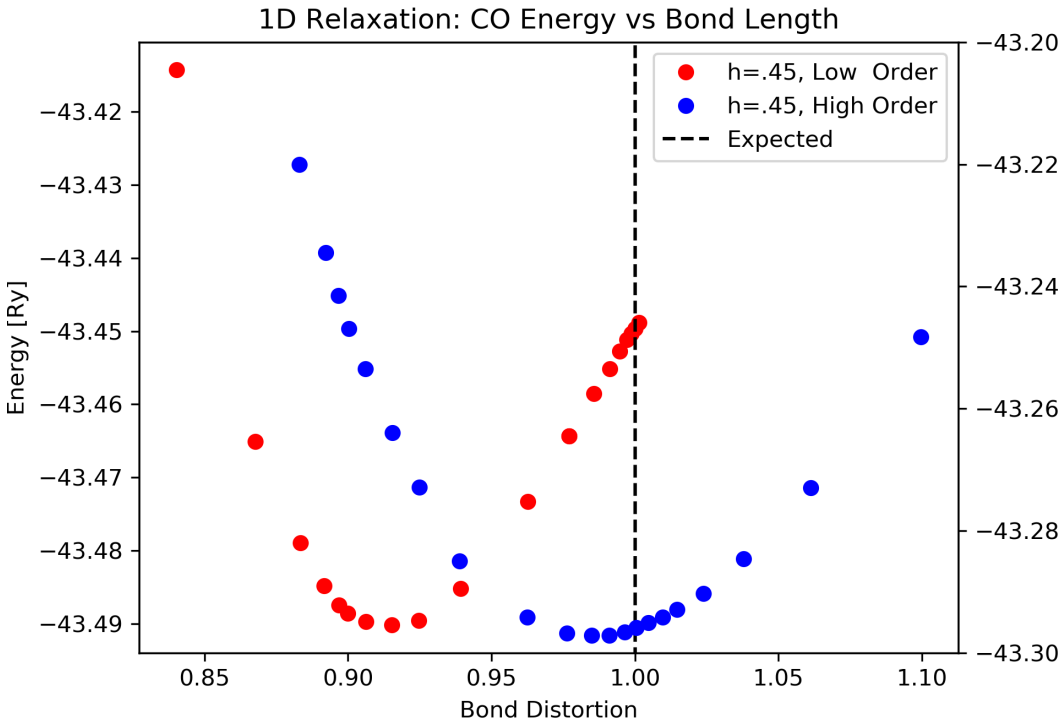

Figure 4.9: Carbon dioxide in an manual relaxation energy minimization attempt. We note that the application of higher order stencils drastically improved the agreement with the correct bond length at higher grid spacing.

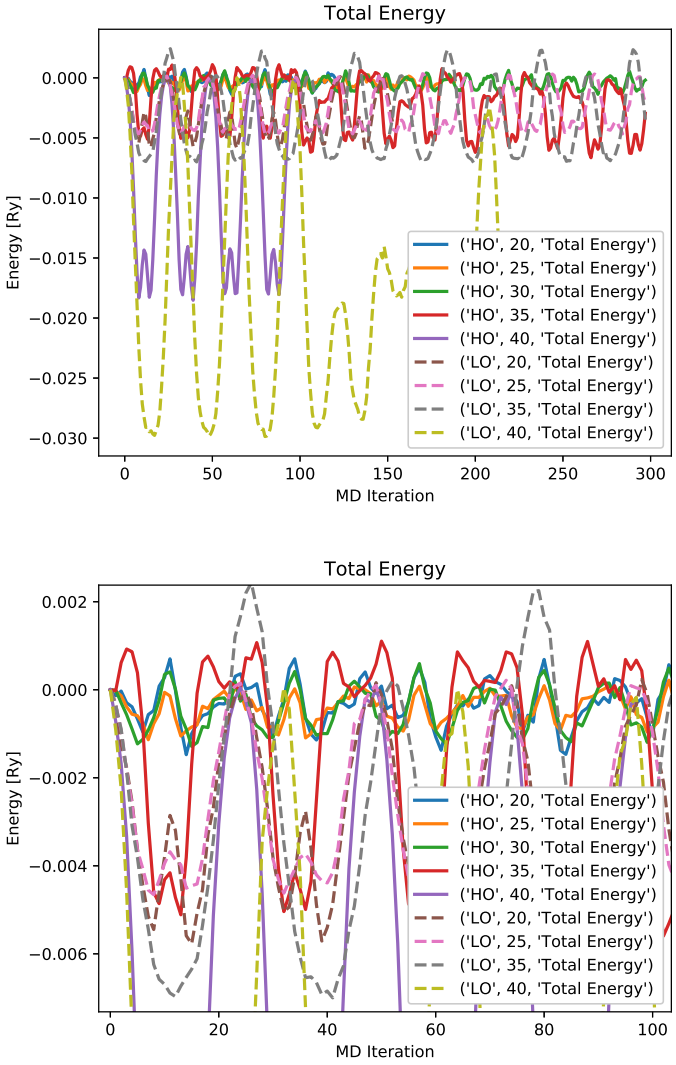

Figure 4.10: The drift of various grid spacing using either the low order (LO) or higher order (HO) integration approaches for O_2. We note that the higher order approach converges to the same sequence of total energies at a much higher grid spacing than the low order approach.

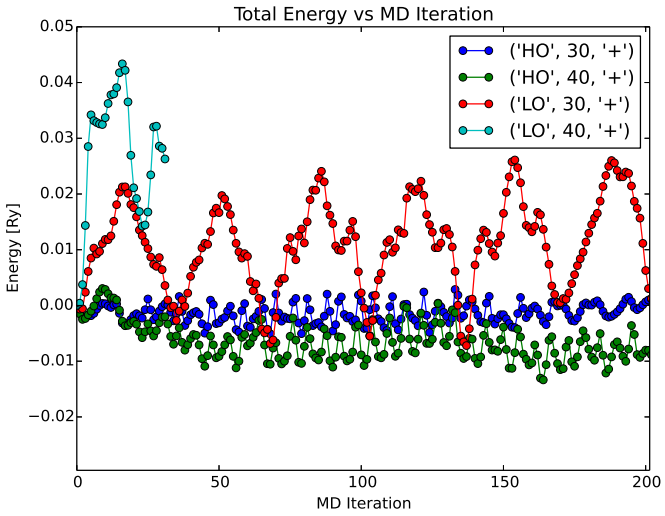

Figure 4.11: The drift of various grid spacing using either the low order (LO) or higher order (HO) integration approaches for C_6H_6. We note that the higher order approach conserves energy better than the low order approach. The LO approach suffers from serious failures at $h = 0.4$

Bibliography

[1] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133, 1965.

[2] P. A. M. Dirac. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 123(792):714–733, 1929.

[3] Tinsley J. Oden. *An Introduction to Mathematical Modeling - A Course In Mechanics*. John Wiley & Sons, Inc, 2011.

[4] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[5] J. C. Slater. A simplification of the hartree-fock method. *Phys. Rev.*, 81:385–390, Feb 1951.

[6] Peter Schwerdtfeger. The pseudopotential approximation in electronic structure theory. *ChemPhysChem*, 12(17):3143–3155, 2011.

[7] James C. Phillips and Leonard Kleinman. New method for calculating wave functions in crystals and molecules. *Phys. Rev.*, 116:287–294, Oct 1959.

[8] D. R. Hamann, M. Schlüter, and C. Chiang. Norm-conserving pseudopotentials. *Phys. Rev. Lett.*, 43:1494–1497, 11 1979.

[9] Leonard Kleinman and D. M. Bylander. Efficacious form for model pseudopotentials. *Physical Review Letters*, 48(20):1425, 1982.

[10] N. Troullier and José Luís Martins. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B*, 43:1993–2006, 1 1991.

[11] Andrew M. Rappe, Karin M. Rabe, Efthimios Kaxiras, and J. D. Joannopoulos. Optimized pseudopotentials. *Phys. Rev. B*, 41:1227–1230, 1 1990.

[12] T.-L. Chan, C. Z. Wang, K. M. Ho, and J. R. Chelikowsky. Efficient first-principles simulation of noncontact atomic force microscopy for structural analysis. *Physical Review Letters*, 102:176101, 2009.

[13] Minjung Kim. *Ab initio simulation methods for the electronic and structural properties of materials applied to molecules, clusters, nanocrystals, and liquids*. PhD thesis, University of Texas at Austin, 2014.

[14] Charles Lena, James Chelikowsky, Ariel Biller, and Leeor Kronik. Advances in real space methods to solve the kohn-sham equation at APS March Meeting 2015. http://meetings.aps.org/link/BAPS.2017.MAR.C7.3, March 2017.

[15] Kevin Gott, Charles Lena, Ariel Biller, Josh Neitzel, Kai-Hsin Liou, Jack Deslippe, and James R Chelikowsky. Scaling and optimization results

of the real-space DFT solver PARSEC on Haswell and KNL systems at IXPUG 2017. https://www.ixpug.org/events/ixpug-2017-US, September 2017.

[16] Grady Schofield. *Computing Accurate Solutions to the Kohn-Sham Problem Quickly in Real Space.* PhD thesis, University of Texas at Austin, 2014.

[17] Charles Lena, James Chelikowsky, Jack Deslippe, Yousef Saad, Chao Yang, and Steven G. Louie. Excited calculations of large scale multi-walled nanotubes using real-space pseudopotential methods at APS March Meeting 2015. http://meetings.aps.org/link/BAPS.2015.MAR.M23.3, March 2015.

[18] Charles Lena, James Chelikowsky, Grady Schofield, Ariel Biller, Leeor Kronik, Yousef Saad, and Jack Deslippe. Scalable real space pseudopotential density functional codes for materials in the exascale regime at APS March Meeting 2015. http://meetings.aps.org/link/BAPS.2016.MAR.S22.1, March 2016.

[19] Alex J. Lee, Minjung Kim, Charles Lena, and James R. Chelikowsky. Mechanical and electronic properties of strained ge nanowires using *ab initio* real-space pseudopotentials. *Phys. Rev. B*, 86:115331, Sep 2012.

[20] Alex J. Lee, Yuki Sakai, Minjung Kim, and James R. Chelikowsky. Repulsive tip tilting as the dominant mechanism for hydrogen bond-like

features in atomic force microscopy imaging. *Applied Physics Letters*, 108(19):193102, 2016.

[21] Yuki Sakai, Alex J Lee, and James Chelikowsky. First-principles atomic force microscopy image simulations with density embedding theory. 16, 04 2016.

[22] N. Scott Bobbitt, Grady Schofield, Charles Lena, and James R. Chelikowsky. High order forces and nonlocal operators in a kohn-sham hamiltonian. *Phys. Chem. Chem. Phys.*, 17:31542–31549, 2015.

[23] Grady Schofield, N. Scott Bobbitt, Charles Lena, and James R. Chelikowsky. Improved forces and nonlocal operators using high- order integration at APS March Meeting 2015. http://meetings.aps.org/link/BAPS.2015.MAR.G24.1, March 2015.

[24] N. Scott Bobbitt. *First principles calculations of Raman spectra for nanostructures and improved high order forces.* PhD thesis, University of Texas at Austin, 2015.

[25] Joshua Neitzel, Grady Schofield, Charles Lena, and James R. Chelikowsky. Improved molecular dynamics calculations using higher-order forces in real space density functional theory. 2017.

[26] J.R. Chelikowsky. The pseudopotential-density functional method applied to nanostructures. *Journal of Physics D: Applied Physics*, 33(8):R33–R50, 2000.

[27] Jorge. Kohanoff. *Electronic Structure Calculations for Solids and Molecules - Theory and Computational Methods*. Cambridge University Press, 2006.

[28] Yukihiro Hasegawa, Jun-Ichi Iwata, Miwako Tsuji, Daisuke Takahashi, Atsushi Oshiyama, Kazuo Minami, Taisuke Boku, Fumiyoshi Shoji, Atsuya Uno, Motoyoshi Kurokawa, Hikaru Inoue, Ikuo Miyoshi, and Mitsuo Yokokawa. First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the k computer. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 1:1–1:11, New York, NY, USA, 2011. ACM.

[29] Yukihiro Hasegawa, Jun-Ichi Iwata, Miwako Tsuji, Daisuke Takahashi, Atsushi Oshiyama, Kazuo Minami, Taisuke Boku, Hikaru Inoue, Yoshito Kitazawa, Ikuo Miyoshi, and Mitsuo Yokokawa. Performance evaluation of ultra-large-scale first-principles electronic structure calculation code on the k computer. *Int. J. High Perform. Comput. Appl.*, 28(3):335–355, 8 2014.

[30] J. Dongarra. Algorithms for future emerging technologies. `https://www.youtube.com/watch?v=TCgHNMezmZ8`, October 2016. Accessed 07/05/17.

[31] Yousef Saad. *NUMERICAL METHODS FOR LARGE EIGENVALUE PROBLEMS*. Society for Industrial and Applied Mathematics, 2nd edition, 2011.

[32] Y. Zhou, Y. Saad, M.L. Tiago, and J.R. Chelikowsky. Self-consistent-field calculations using chebyshev-filtered subspace iteration. *Journal of Computational Physics*, 219:172–184, 2006.

[33] T. Auckenthaler, V. Blum, H.-J. Bungartz, T. Huckle, R. Johanni, L. Krämer, B. Lang, H. Lederer, and P. R. Willems. Parallel solution of partial symmetric eigenvalue problems from electronic structure calculations. *Parallel Computing*, 37:783, Sep 2011.

[34] Andreas Marek, Volker Blum, Rainer Johanni, Ville Havu, Bruno Lang, Thomas Auckenthaler, Alexander Heinecke, Hans-Joachim Bungartz, and Hermann Lederer. The elpa library - scalable parallel eigenvalue solutions for electronic structure theory and computational science. *The Journal of Physics: Condensed Matter*, 26:213201, 2014.

[35] J. Dongarra. The road to exascale and legacy software for dense linear algebra. `https://www.youtube.com/watch?v=EAlJ31OXoSY`, Jun 2017. Accessed 07/06/17.

[36] Mark Frederick Hoemmen. *Communication-avoiding Krylov subspace methods*. PhD thesis, EECS Department, University of California, Berkeley, Apr 2010.

[37] Edgar Solomonik, Erin Carson, Nicholas Knight, and James Demmel. Tradeoffs between synchronization, communication, and work in parallel linear algebra computations. Technical Report UCB/EECS-2014-8, EECS Department, University of California, Berkeley, Jan 2014.

[38] Erin Carson. *Communication-Avoiding Krylov Subspace Methods in Theory and Practice*. PhD thesis, EECS Department, University of California, Berkeley, Aug 2015.

[39] Jochen Alber and Rolf Niedermeier. *On Multi-dimensional Hilbert Indexings*, pages 329–339. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[40] M. Parashar and J. C. Brown. Programming abstraction for parallel adaptive mesh-refinement. *TICAM, University of Texas Austin*, page 4, 1995.

[41] Vincent Michaud-Rioux, Lei Zhang, and Hong Guo. Rescu: A real space electronic structure method. *Journal of Computational Physics*, 307(Supplement C):593 – 613, 2016.

[42] Michael A. Heroux, Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq, Kevin R. Long, Roger P. Pawlowski, Eric T. Phipps, Andrew G. Salinger, Heidi K. Thornquist, Ray S. Tuminaro, James M. Willenbring, Alan Williams, and Kendall S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

[43] Swarnava Ghosh and Phanish Suryanarayana. Sparc: Accurate and efficient finite-difference formulation and parallel implementation of density functional theory: Isolated clusters. *Computer Physics Communications*, 212(Supplement C):189 – 204, 2017.

[44] Grady Schofield, James R. Chelikowsky, and Yousef Saad. A spectrum slicing method for the Kohn-Sham problem. *Computer Physics Communications*, 183:497, 2012.

[45] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. Parallel self-consistent-field calculations via chebyshev-filtered subspace acceleration. *Phys. Rev. E*, 74:066704, 12 2006.

[46] Yunkai Zhou, James R. Chelikowsky, and Yousef Saad. Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the kohn-sham equation. *Journal of Computational Physics*, 274:770 – 782, 2014.

[47] Yunkai Zhou and Yousef Saad. A chebyshev-davidson algorithm for large symmetric eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 29(3):954–971, 2007.

[48] Azzam Haidar, Piotr Luszczek, Jakub Kurzak, and Jack Dongarra. An improved parallel singular value algorithm and its implementation for multicore hardware. `http://www.netlib.org/lapack/lawnspdf/lawn283.pdf`, 2013.

[49] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[50] Jack Poulson, Bryan Marker, Robert A. van de Geijn, Jeff R. Hammond, and Nichols A. Romero. Elemental: A new framework for distributed memory dense matrix computations. *ACM Transactions on Mathematical Software*, 39(2):13:1–13:24, 2013.

[51] Nersc exascale scientific application program, Jun 2016.

[52] J. R. Chelikowsky, T.-L. Chan, M. M. G. Alemany, and G. Dalpian. Computational studies of doped nanostructures. *Rep. Prog. Phys.*, 74:046501, 2011.

[53] Jian-Yih Cheng, Brandon L. Fisher, Nathan P. Guisinger, and Carmen M. Lilley. Atomically manufactured nickel–silicon quantum dots displaying robust resonant tunneling and negative differential resistance. *npj Quantum Materials*, 2(1):25, 2017.

[54] R. Car and M. Parrinello. Structural, dymanical, and electronic properties of amorphous silicon: An ab initio molecular-dynamics study. *Physical Review Letters*, 60:204, 1988.

[55] R. Biswas and D. R. Hamann. Simulated annealing of silicon atom clusters in langevin molecular dynamics. *Physical Review B*, 34:895, 1986.

[56] Tonya M Atkins, Maja C Cassidy, Menyoung Lee, Shreyashi Ganguly, Charles M Marcus, and Susan M Kauzlarich. Synthesis of long-t(1) sili-

con nanoparticles for hyperpolarized (29)si magnetic resonance imaging. *ACS nano*, 7(2):1609–1617, 02 2013.

[57] R.P. Feynman. Forces in molecules. *Physical Review*, 56:340, 1939.

[58] Alberto Pérez, F Javier Luque, and Modesto Orozco. Frontiers in molecular dynamics simulations of dna. *Accounts of chemical research*, 45(2):196–205, 2011.

[59] Run Long and Oleg V Prezhdo. Ab initio nonadiabatic molecular dynamics of the ultrafast electron injection from a pbse quantum dot into the tio2 surface. *Journal of the American Chemical Society*, 133(47):19240–19249, 2011.

[60] Geoffrey R Luckhurst and Carlo Alberto Veracini. *The Molecular Dynamics of Liquid Crystals*, volume 431. Springer Science & Business Media, 2012.

[61] Adarsh Shekhar, Ken-ichi Nomura, Rajiv K Kalia, Aiichiro Nakano, and Priya Vashishta. Nanobubble collapse on a silica surface in water: Billion-atom reactive molecular dynamics simulations. *Physical review letters*, 111(18):184503, 2013.

[62] J Postma, Ronnie Hoekstra, AGGM Tielens, and Thomas Schlathölter. A molecular dynamics study on slow ion interactions with the polycyclic aromatic hydrocarbon molecule anthracene. *The Astrophysical Journal*, 783(1):61, 2014.

[63] Amy Khoo, Minjug Kim, Grady Schofield, and James R. Chelikowsky. *Ab initio* molecular dynamics simulations using a Chebyshev-filtered subspace iteration technique. *Physical Review B*, 82:064201, 2010.

[64] KH Khoo, T-L Chan, M Kim, and James R Chelikowsky. Ab initio molecular dynamics simulations of molten $al_{1-x}si_x$ alloys. *Physical Review B*, 84(21):214203, 2011.

[65] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.

[66] SÃžren Toxvaerd, Ole J. Heilmann, and Jeppe C. Dyre. Energy conservation in molecular dynamics simulations of classical systems. *The Journal of Chemical Physics*, 136(22):–, 2012.

[67] M. J. Cawkwell and Anders M. N. Niklasson. Energy conserving, linear scaling born-oppenheimer molecular dynamics. *The Journal of Chemical Physics*, 137(13):–, 2012.